

POLITECNICO DI MILANO  
FACOLTÀ DI INGEGNERIA DEI SISTEMI  
CORSO DI STUDI IN INGEGNERIA MATEMATICA



TESI DI LAUREA MAGISTRALE

**OTTIMIZZAZIONE DI FORMA  
PER FLUSSI VISCOSI TRIDIMENSIONALI  
IN GEOMETRIE CARDIOVASCOLARI**

Relatore: Prof. Sandro SALSA  
Correlatore: Dott. Ing. Andrea MANZONI

Francesco BALLARIN  
matricola 749944

ANNO ACCADEMICO 2010/2011

# Indice

<b>Sommario</b>	<b>vii</b>
<b>Abstract</b>	<b>viii</b>
<b>Introduzione</b>	<b>1</b>
Problemi di controllo ottimo e problemi di ottimizzazione di forma . . . . .	1
Struttura della Tesi . . . . .	2
<b>1 Motivazioni</b>	<b>3</b>
1.1 Patologie coronariche e principali trattamenti . . . . .	3
1.2 Modellistica matematica per l'emodinamica . . . . .	5
1.3 Ottimizzazione di forma di un bypass aorto-coronarico . . . . .	7
<b>2 Formulazione astratta di un problema di ottimizzazione di forma</b>	<b>11</b>
2.1 Problemi di controllo ottimo per EDP . . . . .	11
2.1.1 Alcuni risultati per la minimizzazione di funzionali . . . . .	12
2.1.2 Controllo ottimo per problemi ellittici: approccio alla Lions . . . . .	14
2.1.3 Controllo ottimo per problemi ellittici: il metodo dei moltiplicatori di Lagrange	15
2.2 Problemi di ottimizzazione di forma . . . . .	18
2.2.1 Topologie ed una proprietà di compattezza . . . . .	20
2.2.2 Domini uniformemente regolari . . . . .	21
2.2.3 Semicontinuità inferiore del funzionale costo . . . . .	22
2.2.4 Un esempio: equazione di Poisson . . . . .	23
2.3 Domini: descrizione e deformazione . . . . .	26
2.3.1 La mappa <i>Free Form Deformation</i> . . . . .	27
2.3.2 Derivata di forma . . . . .	32
2.3.3 Calcolo della <i>shape derivative</i> per l'equazione di Poisson . . . . .	34
2.4 Condizioni di ottimalità . . . . .	35
2.4.1 L'approccio basato sul problema aggiunto . . . . .	35
2.4.2 Metodo dei moltiplicatori di Lagrange . . . . .	38
<b>3 Ottimizzazione di forma per problemi di fluidodinamica</b>	<b>42</b>
3.1 Descrizione del modello fluidodinamico . . . . .	44
3.2 Problema di Stokes stazionario . . . . .	45
3.2.1 Formulazione debole del problema di stato . . . . .	46
3.2.2 Un risultato di esistenza per il problema di ottimizzazione di forma . . . . .	47
3.2.3 Un risultato di simmetria per la forma ottima . . . . .	49
3.2.4 Calcolo della <i>shape derivative</i> . . . . .	50
3.2.5 Formulazione del problema aggiunto . . . . .	52
3.2.6 Risoluzione esplicita del problema aggiunto per forzante e dato di Neumann nulli . . . . .	55

3.3	Problema di Stokes non stazionario . . . . .	55
3.3.1	Formulazione debole del problema di stato . . . . .	56
3.3.2	Un risultato di esistenza per il problema di ottimizzazione di forma . . . . .	57
3.3.3	Calcolo della <i>shape derivative</i> . . . . .	60
3.3.4	Formulazione del problema aggiunto . . . . .	61
<b>4</b>	<b>Metodi numerici per problemi di ottimizzazione di forma</b>	<b>68</b>
4.1	Schema iterativo per il calcolo di una forma ottima . . . . .	68
4.2	Approssimazione del problema di stato e del problema aggiunto . . . . .	69
4.2.1	Problema di Stokes stazionario . . . . .	69
4.2.2	Problema di Stokes non stazionario . . . . .	71
4.3	Ottimizzazione di forma mediante <i>local boundary variation</i> . . . . .	71
4.3.1	Algoritmo di ottimizzazione numerica . . . . .	72
4.3.2	Gestione del vincolo di volume . . . . .	73
4.4	Ottimizzazione di forma mediante parametrizzazione con mappa <i>Free Form Deformation</i> . . . . .	74
4.4.1	Condizioni di ottimalità rispetto ai parametri della mappa <i>Free Form Deformation</i> . . . . .	74
4.4.2	Algoritmo di ottimizzazione numerica . . . . .	76
4.4.3	Gestione del vincolo di volume . . . . .	77
4.4.4	Presentazione della libreria C++ . . . . .	78
<b>5</b>	<b>Simulazioni numeriche</b>	<b>80</b>
5.1	Due casi test con equazione di Poisson . . . . .	80
5.1.1	Caso test stazionario . . . . .	80
5.1.2	Caso test non stazionario . . . . .	85
5.2	Corpo immerso in un flusso di Stokes . . . . .	87
5.2.1	Problema di Stokes stazionario . . . . .	88
5.2.2	Problema di Stokes non stazionario . . . . .	91
5.3	Bypass aorto-coronarico . . . . .	96
5.3.1	Problema di Stokes stazionario . . . . .	98
5.3.2	Problema di Stokes non stazionario . . . . .	110
5.3.3	Validazione con problema di Navier-Stokes non stazionario . . . . .	121
	<b>Conclusioni</b>	<b>146</b>
<b>A</b>	<b>Implementazione in C++</b>	<b>148</b>
A.1	La mappa <i>Free Form Deformation</i> . . . . .	148
A.1.1	Classe Displacement . . . . .	148
A.1.2	Classe ControlPoint . . . . .	149
A.1.3	Classe FFDReferenceMesh<RegionMesh> . . . . .	149
A.1.4	Classe FFDMeshData . . . . .	150
A.1.5	Classe Bernstein1D . . . . .	151
A.1.6	Classe FreeFormDeformation<RegionMesh> . . . . .	151
A.2	Metodo del gradiente con proiezione . . . . .	154
A.2.1	Classe Function . . . . .	154
A.2.2	Classi per la gestione di problemi vincolati . . . . .	154
A.2.3	Classe Exporter . . . . .	156
A.2.4	Classe StoppingCriteria . . . . .	156
A.2.5	Classe BacktrackingData . . . . .	157
A.2.6	Classe ProjectionMethodData . . . . .	157
A.2.7	Classe ProjectionMethod . . . . .	158

---

A.3	Generico problema di ottimizzazione di forma . . . . .	161
A.3.1	Classe ShapeOptimizationData . . . . .	163
A.3.2	Classi per la memorizzazione delle soluzioni dei problemi di stato ed aggiunto	164
A.3.3	Classi per la gestione della dipendenza temporale . . . . .	164
A.3.4	Classe ShapeOptimizationProblem<ShapeParametrization> . . . . .	165
A.3.5	Classe FFDShapeOptimizationProblem<RegionMesh> . . . . .	168
A.3.6	Classe FFDPProjectionMethodShapeOptimizationProblem<Re- gionMesh> . . . . .	173
A.3.7	Classe FFDPProjectionMethodShapeOptimizationProblem_Deco- ratorBase<RegionMesh> . . . . .	178
A.3.8	Classe AddVolumeConstraintDecorator<RegionMesh> . . . . .	178
A.4	Un esempio: implementazione del caso test della Sezione 5.1.1 . . . . .	178
	<b>Ringraziamenti</b>	<b>181</b>
	<b>Bibliografia</b>	<b>182</b>



# Elenco delle figure

1.1	Cuore, aorta e principali arterie coronarie. . . . .	4
1.2	Stenosi coronarica. . . . .	4
1.3	Bypass aorto-coronarico. . . . .	5
1.4	Modelli semplificati. . . . .	7
1.5	Alcune possibili geometrie semplificate di un bypass aorto-coronarico. . . . .	8
2.1	Proprietà dell' $\varepsilon$ -cono. . . . .	21
2.2	Mappa <i>perturbazione dell'identità</i> . . . . .	26
2.3	Esempi di deformazione secondo la mappa <i>Free Form Deformation</i> . . . . .	27
2.4	Diagramma della costruzione della mappa <i>Free Form Deformation</i> . . . . .	29
3.1	Geometrie. . . . .	43
4.1	Diagramma della costruzione della mappa $\mathfrak{T}(\mathbf{x}_\mu, t; \boldsymbol{\mu}, \boldsymbol{\eta})$ associata alla <i>Free Form Deformation</i> . . . . .	75
5.1	Configurazione dei punti di controllo. I punti in grigio sono vincolati, quelli in rosa hanno tre gradi di libertà. . . . .	82
5.2	Direzioni di discesa all'iterazione iniziale (simulazione 4). . . . .	82
5.3	Forma iniziale e finale per la simulazione 4. . . . .	82
5.4	Forma iniziale e finale per la simulazione 5. . . . .	83
5.5	Storia di convergenza delle simulazioni 3 e 4. . . . .	84
5.6	Dominio di riferimento e punti di controllo della mappa <i>Free Form Deformation</i> : il dominio di riferimento è dato dall'intersezione tra il complementare della palla (indicata in blu) ed il <i>bounding box</i> $D$ (indicato in azzurro). I punti in grigio sono vincolati, quelli in rosa hanno un grado di libertà attivo nella direzione normale alla superficie della sfera. . . . .	87
5.7	Sezione del profilo ottimo individuato da Pironneau in [Pir73], con $\alpha = \beta = 120^\circ$ . . . . .	88
5.8	Velocità nella sezione longitudinale. . . . .	89
5.9	<i>Streamlines</i> . . . . .	90
5.10	Confronto con i risultati di Pironneau [Pir73] e Ogawa [OK03]. . . . .	90
5.11	Velocità di picco per la sezione di inlet. . . . .	91
5.12	Confronto tra la forma ottima del caso stazionario (in blu) e quella del caso non stazionario (in rosso). . . . .	93
5.13	Velocità nella sezione longitudinale a diversi istanti temporali. . . . .	94
5.14	<i>Streamlines</i> a diversi istanti temporali. . . . .	95
5.15	Configurazione indeformata. . . . .	97
5.16	Simulazione <i>a</i> . . . . .	102
5.17	Simulazione <i>b</i> . . . . .	103
5.18	Simulazione <i>c</i> . . . . .	104
5.19	Simulazioni $d'$ e $d''$ . . . . .	105

---

5.20	Velocità nella sezione longitudinale. . . . .	106
5.21	Profili di velocità in diverse sezioni. . . . .	107
5.22	<i>Streamlines</i> . . . . .	108
5.23	Modulo della sforzo di taglio a parete nella regione a valle della stenosi. . . . .	109
5.24	Velocità di picco per la sezione di inlet. . . . .	110
5.25	Simulazione <i>e</i> . . . . .	113
5.26	Simulazione <i>f</i> . . . . .	114
5.27	Velocità nella sezione longitudinale a diversi istanti temporali. . . . .	115
5.28	Profili di velocità su varie sezioni a diversi istanti temporali. . . . .	116
5.29	<i>Streamlines</i> a diversi istanti temporali. . . . .	118
5.30	Modulo della sforzo di taglio a parete nella regione a valle della stenosi, a diversi istanti temporali. . . . .	119
5.31	Confronto con alcuni risultati tratti da [WZJ <sup>+</sup> 11, SM10]. . . . .	122
5.32	Velocità nella sezione longitudinale a diversi istanti temporali. . . . .	125
5.33	Profili di velocità su varie sezioni a diversi istanti temporali. . . . .	129
5.34	<i>Streamlines</i> a diversi istanti temporali. . . . .	133
5.35	Modulo della vorticità a diversi istanti temporali. . . . .	137
5.36	Modulo della sforzo di taglio a parete nella regione a valle della stenosi, a diversi istanti temporali. . . . .	141
5.37	<i>Oscillating shear index</i> . . . . .	145
A.1	<i>Collaboration diagram</i> della classe <code>FreeFormDeformation&lt;RegionMesh&gt;</code> . . . . .	155
A.2	<i>Collaboration diagram</i> della classe <code>ProjectionMethod</code> . . . . .	162
A.3	<i>Collaboration diagram</i> della classe <code>FFDProjectionMethodShapeOptimizationProblem&lt;RegionMesh&gt;</code> . . . . .	177

# Elenco delle tabelle

5.1	Risultati del caso test stazionario. . . . .	81
5.2	Risultati del caso test non stazionario. . . . .	86
5.3	Risultati dell'ottimizzazione di forma di un bypass aorto-coronarico, con flusso stazionario. . . . .	99
5.4	Risultati dell'ottimizzazione di forma di un bypass aorto-coronarico, con flusso non stazionario. . . . .	111
A.1	Analisi di scalabilità. . . . .	180

# Sommario

Nel corso degli ultimi decenni i problemi di **ottimizzazione di forma** hanno acquistato un'attenzione sempre crescente, soprattutto grazie alle numerose applicazioni in fisica o ingegneria nelle quali, per migliorare il comportamento di un sistema, occorre ottimizzare la forma stessa di un corpo o di un dispositivo. In questa Tesi vengono presentati metodi analitici e numerici per risolvere problemi di ottimizzazione di forma in ambito fluidodinamico, per flussi tridimensionali (a basso numero di Reynolds) descritti dalle equazioni di Stokes stazionarie e non stazionarie, con applicazione a problemi di natura emodinamica, quali ad esempio il *design* ottimo di un bypass aorto-coronarico.

Dal punto di vista **teorico**, viene presentata anzitutto un'analisi astratta di buona posizione, con particolare attenzione alle ipotesi di regolarità sulle forme ammissibili. In vista di una riduzione della complessità geometrica, le deformazioni dei domini vengono descritte mediante trasformazioni di tipo *perturbazione dell'identità*; si ricavano poi le *condizioni di ottimalità* introducendo un problema aggiunto e utilizzando il metodo dei moltiplicatori di Lagrange.

Per quanto riguarda gli aspetti **computazionali**, vengono descritti e implementati alcuni metodi numerici per la risoluzione approssimata di problemi di ottimizzazione di forma, sviluppati estendendo la libreria *LifeV*. In particolare, vengono impiegate mappe parametriche di tipo *Free Form Deformation*: esse permettono di descrivere una vasta classe di forme ammissibili in modo computazionalmente efficiente, operando perturbazioni dell'identità mediante un numero ridotto di parametri, che assumono il ruolo di variabili di ottimizzazione.

La principale **applicazione** discussa in questa Tesi riguarda l'ottimizzazione di forma di un bypass aorto-coronarico, della quale viene riportata una dettagliata analisi e numerosi confronti con risultati presenti in letteratura.

**Parole chiave:** ottimizzazione di forma; flussi di Stokes tridimensionali (stazionari e non stazionari); bypass aorto-coronarico; *Free Form Deformation*.

# Abstract

**Shape optimization** problems have become extremely popular during the last decades, mainly because of several advanced applications in physics or engineering, in which the optimal design of a particular device greatly enhances the efficiency or the mechanical behavior of a system. The goal of this Thesis is to address some analytical and numerical methods for the solution of shape optimization problems in the framework of computational fluid dynamics, dealing with three-dimensional flows (at low Reynolds numbers) modelled by steady or unsteady Stokes equations. In particular, these methods have been applied to problems arising from haemodynamics, such as the optimal design of an aorto-coronary bypass graft.

From the **theoretical** point of view, several tools for the analysis of shape optimization problems are introduced, focusing on the regularity assumptions which ensure the well-posedness of the problem. In view of a reduction of the geometrical complexity, domain deformations are described introducing suitable *identity perturbation* maps; hence, the *optimality conditions* are derived by means of an adjoint-based procedure, exploiting the Lagrange multipliers method.

Concerning the **computational** aspects, a numerical procedure for the discrete solution of shape optimization problems is described; moreover, a new solver has been developed extending the capabilities of the *LifeV* finite element library. In particular, *Free Form Deformation* maps have been employed, since they provide a flexible and efficient tool for representing global and smooth deformations acting on a small number of design parameters, which play the role of optimization variables.

The main **application** in this Thesis deals with the optimal shape design of aorto-coronary bypass grafts, for which a detailed analysis and several comparisons with previous results are provided.

**Keywords:** shape optimization; three-dimensional steady and unsteady Stokes flows; aorto-coronary bypass; *Free Form Deformation*.

# Introduzione

## Problemi di controllo ottimo e problemi di ottimizzazione di forma

Numerosi settori dell'ingegneria richiedono lo studio e la risoluzione di un problema di controllo: si pensi ad esempio alla gestione di un impianto di produzione industriale in cui, integrando opportunamente le varie unità, ci si pone l'obiettivo di convertire, più economicamente possibile, certe materie prime in un prodotto, garantendo che l'impianto soddisfi sempre una serie di condizioni specificate durante il progetto (vincoli operativi sulle macchine, specifiche di produzione, aderenza a regolamentazioni giuridiche ed ambientali, sicurezza degli operatori, ...). Un secondo esempio può essere quello della gestione dell'impianto di climatizzazione di un edificio in cui, dopo aver misurato la temperatura in ciascuna stanza, sia necessario attivare o regolare selettivamente alcuni terminali (radiatori, termoconvettori, ...) fino a raggiungere la temperatura desiderata.

La struttura comune a ciascuno di questi casi è la presenza di un *sistema fisico* sul quale è necessario intervenire con un opportuno *controllo* in modo da ottenere un *comportamento desiderato*, rispettando determinati *vincoli*.

Tuttavia, dato che solitamente una descrizione esaustiva del sistema fisico è troppo complessa, ci si limita a sostituirlo con un opportuno *modello*, il quale, a seconda dell'applicazione in esame e del livello di dettaglio da essa richiesto, può essere espresso come un sistema algebrico, un sistema dinamico, un processo stocastico, un problema differenziale alle derivate ordinarie o alle derivate parziali. È possibile infine estrarre dal modello alcune *variabili di osservazione*, che guideranno la scelta del controllo da applicare al sistema.

Gli obiettivi della teoria del controllo ottimo possono essere riassunti come segue:

1. dimostrare l'esistenza di almeno un controllo che garantisca il comportamento desiderato;
2. caratterizzare le proprietà del controllo ottimo;
3. sfruttare questa caratterizzazione per ricavare analiticamente o con opportuni metodi numerici il "valore" del controllo ottimo.

In questa Tesi ci si occuperà in particolare di modelli descritti da equazioni alle derivate parziali e, soprattutto, di problemi di *ottimizzazione di forma*, cioè particolari problemi di controllo ottimo per EDP in cui la variabile di controllo è il dominio stesso su cui sono definite le equazioni che costituiscono il modello del sistema fisico. Questi problemi occupano uno spazio considerevole in numerosi settori dell'ingegneria: storicamente, il primo contesto industriale che ha manifestato la necessità di problemi di questo tipo è quello dell'aeronautica, con il problema dell'ottimizzazione della forma di un profilo alare [HMV74, HH78, Jam95, Jam88], per il quale una diminuzione di pochi punti percentuali del *drag* può comportare un notevole risparmio in termini di carburante. Altre applicazioni sono studiate in ingegneria strutturale (problemi di elasticità per travi e membrane, [Hem73, Kir81, Pra74, Roz89, RBK95]), ingegneria navale (ottimizzazione delle componenti di imbarcazioni da competizione [PQ05, Man08, LPQR11]) o nel *design* di dispositivi biomedici (in particolare *graft* cardiovascolari [QR03, Roz05a, AQR07, LAK97, MFT08, PLN<sup>+</sup>10, SM10]).

## Struttura della Tesi

Il CAPITOLO 1 presenterà, prima da un punto di vista medico, le malattie a cui le arterie coronarie possono essere soggette, e per le quali è necessario intervenire con operazioni di bypass aorto-coronarico o di angioplastica coronarica. La fluidodinamica computazionale può aiutare a comprendere maggiormente il comportamento del flusso sanguigno sia nell'arteria occlusa sia dopo l'innesto di un bypass o di uno stent. In questo lavoro di Tesi vengono discussi alcuni problemi di ottimizzazione di forma in ambito emodinamico, con l'obiettivo finale di determinare una configurazione ottima per l'innesto di un bypass aorto-coronarico.

Nel CAPITOLO 2 verrà descritta la formulazione astratta di un problema di ottimizzazione di forma, inserita nell'ambito più generale dei problemi di controllo ottimo per EDP. Verrà presentato un risultato astratto che garantisce l'esistenza di almeno una forma ottima, discutendo in particolare le ipotesi necessarie. Dopo aver mostrato la buona posizione del problema, verrà illustrato come introdurre un opportuno problema aggiunto per ricavare le condizioni che caratterizzano la forma ottima: esse sono basate sull'introduzione della famiglia di mappe di *perturbazione dell'identità*, che permette di definire la derivazione rispetto alla forma di integrali su domini variabili. Verrà presentata una particolare mappa, detta *Free Form Deformation*, mostrando che essa appartiene alla famiglia delle perturbazioni dell'identità; questo tipo di mappa sarà utilizzata inoltre per il trattamento della forma e delle relative deformazioni nelle simulazioni numeriche, in modo da ottenere una procedura di ottimizzazione computazionalmente efficiente.

Il CAPITOLO 3 sarà dedicato a problemi di ottimizzazione di forma in ambito fluidodinamico, e in particolare all'analisi del problema di minimizzazione dell'energia dissipata, sia per un corpo immerso in un fluido viscoso incomprimibile, sia per la geometria di un bypass aorto-coronarico. Come problema di stato verranno considerate sia le equazioni di Stokes stazionarie, sia le equazioni di Stokes non stazionarie: in entrambi i casi, dopo aver ricavato la forma debole del problema di stato, si dimostrerà l'esistenza di almeno una forma ottima, ricavando in particolare la continuità della soluzione del problema di stato rispetto al dominio. Infine, la forma ottima verrà caratterizzata introducendo un opportuno problema aggiunto.

Il CAPITOLO 4 presenterà alcuni possibili metodi di approssimazione numerica per problemi di ottimizzazione di forma. Dopo aver introdotto uno schema iterativo generale, ne verranno presentate le due principali componenti: la discretizzazione ad elementi finiti del problema di stato e l'aggiornamento della forma tra una iterazione e la successiva. In particolare, verranno presentate due possibili strategie per la parametrizzazione della forma, basate su *local boundary variation* e su mappe di tipo *Free Form Deformation*. Si mostrerà inoltre che quest'ultima parametrizzazione è particolarmente conveniente da un punto di vista computazionale, in quanto permette di ridurre la dimensionalità del problema di ottimizzazione.

A partire dalla libreria *LifeV*, è stato implementato un risolutore in C++ per un generico problema di ottimizzazione di forma con parametrizzazione *Free Form Deformation*, presentato in dettaglio nell'APPENDICE A. Le simulazioni numeriche condotte per mezzo di questo risolutore verranno illustrate nel CAPITOLO 5. Dopo aver mostrato due casi test con problema di stato dato dall'equazione di Poisson, verrà discussa l'ottimizzazione della forma di un corpo immerso in un flusso di Stokes, confrontando anche le proprietà della parametrizzazione mediante mappa *Free Form Deformation* con risultati ottenuti mediante tecniche di *local boundary variation*. Infine, verrà presentata l'applicazione più importante di questa Tesi, cioè la ricerca di un *design* ottimale di un bypass aorto-coronarico, riportandone una dettagliata analisi e numerosi confronti con risultati presenti in letteratura.

## Capitolo 1

# Motivazioni

### 1.1 Patologie coronariche e principali trattamenti

Le patologie del sistema cardiovascolare rappresentano, attualmente, la principale causa di mortalità nel nostro Paese: circa il 45% di tutti i decessi sono, infatti, attribuibili a malattie cardiovascolari. In particolare, dei 250.000 casi di decesso annuale per patologie cardiocircolatorie, i due terzi sono imputabili a malattie coronariche<sup>1</sup>. La circolazione coronarica è costituita dai vasi sanguigni che portano sangue al muscolo cardiaco. Sebbene durante il ciclo cardiaco, dato dal susseguirsi del movimento di sistole e diastole, gli atri e i ventricoli si riempiano di sangue e, successivamente, si svuotano, il miocardio (tessuto muscolare del cuore) necessita di una propria rete vascolare, in grado di irrorarlo e sopperire al suo fabbisogno metabolico. Questa rete è costituita dalle arterie coronarie, che sono i vasi che portano al miocardio sangue ricco di ossigeno, e dalle vene cardiache, che rimuovono dal muscolo cardiaco il sangue ricco di anidride carbonica. L'anatomia delle arterie coronarie è molto variabile da individuo a individuo; in generale si individuano due arterie principali – le arterie coronarie destra e sinistra – e, talvolta, una terza arteria, detta coronaria posteriore. Entrambe le arterie coronarie principali hanno origini dal primo tratto dell'aorta (si veda la Figura 1.1); in particolare, l'arteria coronaria destra, passando nel solco coronarico alla base del ventricolo destro, raggiunge il solco interventricolare posteriore sulla faccia diaframmatica del cuore. L'arteria coronaria sinistra è abbastanza breve, perché a pochi millimetri dalla sua origine si divide in un ramo anteriore discendente, che scende verso il basso nel solco interventricolare anteriore, ed in un ramo circonflesso, che raggiunge il solco coronarico intorno alla base del ventricolo sinistro. Da queste arterie coronarie principali, che si trovano sulla superficie dell'epicardio (rivestimento della superficie esterna del cuore), si suddividono ulteriori ramificazioni più sottili, che penetrano nel miocardio e lo irrorano di sangue.

Dato che le arterie coronarie devono sopperire all'intero fabbisogno metabolico del cuore, una limitazione del flusso sanguigno in queste arterie provoca disturbi al funzionamento dell'intero sistema cardiocircolatorio, in misura proporzionale all'entità della diminuzione di trasporto di ossigeno al miocardio.

Una delle principali cause della compromissione del flusso coronarico è dovuta alla presenza di una occlusione (o *stenosi*). La causa più comune delle stenosi coronariche è l'*aterosclerosi*: inizialmente indotta da un deposito di lipidi sulla parete del vaso, questa patologia provoca un limitato flusso sanguigno attraverso i vasi coronarici, che l'organismo tenta di compensare con una dilatazione dei vasi a valle. Nella fase iniziale di questa patologia si assiste alla formazione di una placca (detta *placca aterosclerotica*, mostrata in Figura 1.2(a)), costituita da molecole di lipidi fuse tra loro ed infiltrate di altre sostanze, come il colesterolo. All'aumentare dello spessore della placca, la patologia non comporta solo un progressivo restringimento del diametro vascolare, ma anche una limitazione delle capacità elastiche del vaso: nelle fasi più acute la capacità dei vasi di

---

<sup>1</sup>Dati tratti da [PF08]. La seguente breve descrizione fisiologica e medica delle arterie coronarie è tratta da [RP76, HOWF09], a cui si rimanda per ulteriori dettagli.



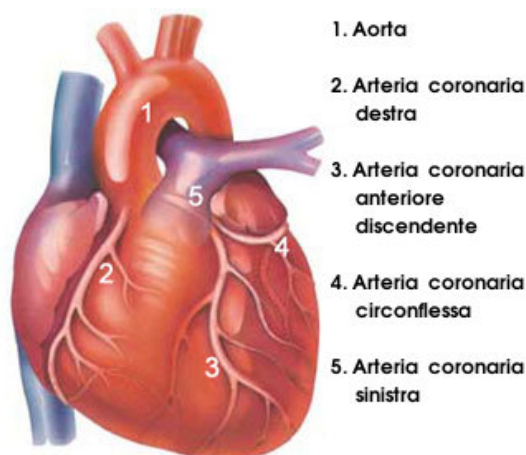
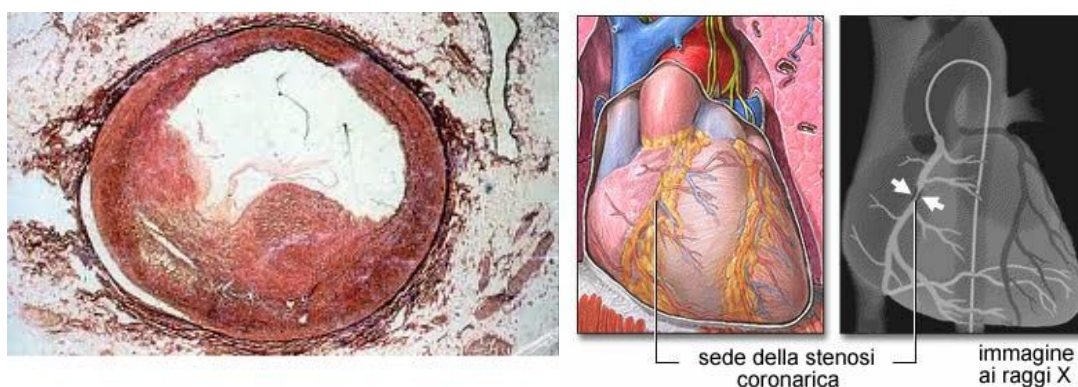


Figura 1.1: Cuore, aorta e principali arterie coronarie.



(a) Placca aterosclerotica.

(b) Sede della stenosi coronarica.

Figura 1.2: Stenosi coronarica.

dilatarsi risulta insufficiente, la richiesta di ossigeno da parte del miocardio supera rapidamente la quantità fornita dal flusso coronarico e si può dunque verificare una *ischemia del miocardio*, cioè il miocardio riceve una quantità insufficiente di sangue. L'ischemia è una patologia transitoria e reversibile, i cui sintomi possono essere alleviati con l'assunzione di farmaci; tuttavia, nei casi più gravi, la rottura della placca aterosclerotica e la conseguente formazione di trombi o emboli può comportare una condizione ischemica prolungata nel tempo e provocare l'*infarto del miocardio* (necrosi del tessuto cardiaco).

La manifestazione più comune di un'insufficienza del flusso coronarico è l'*angina pectoris*, caratterizzata, soprattutto durante gli sforzi fisici, da un iniziale dolore al petto che può irradiarsi anche al collo e alla braccia. Una *coronarografia* permette di evidenziare la sede della stenosi (Figura 1.2(b)): nei casi più acuti occorre procedere alla rivascolarizzazione mediante *bypass aorto-coronarico* (*coronary artery bypass graft*, CABG) o con un intervento di *angioplastica coronarica* (*percutaneous transluminal coronary angioplasty*, PTCA). Il bypass aorto-coronarico (Figura 1.3) consiste in una procedura chirurgica che prevede l'innesto di un nuovo condotto vascolare (*graft*) per aggirare l'occlusione. Durante l'intervento, il chirurgo apre le arterie coronarie a valle dell'ostruzione e vi sutura un segmento di vena safena (prelevata da una gamba del paziente), la cui estremità a monte della stenosi (o prossimale) viene poi suturata all'aorta ascendente. Fin dagli anni '70-'80 l'uso della vena safena è stato sostituito da quello dell'arteria toracica interna, in

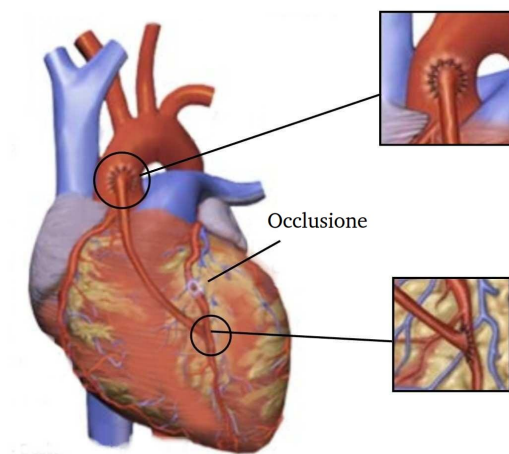


Figura 1.3: Bypass aorto-coronarico.

quanto richiede solo la sutura dell'estremità a valle della stenosi (o distale), più conveniente del precedente perché l'estremità prossimale è già collegata al sistema cardiovascolare tramite l'arteria succlavia: inoltre, le protesi venose hanno un'incidenza di ristrenosi che arriva al 50% in 10 anni, al contrario di quelle arteriose che garantiscono una affidabilità maggiore. In tempi più recenti è stata considerata anche la possibilità di utilizzare innesti di tipo sintetico, ad esempio di politetrafluoroetilene (PTFE): la procedura chirurgica è meno invasiva rispetto all'utilizzo di materiali organici, perché non richiede l'estrazione della vena safena o dell'arteria toracica interna, ma è necessario trattare opportunamente l'innesto per evitare il rigetto.

Vengono solitamente sottoposti all'intervento di bypass aorto-coronarico i pazienti con gravi insufficienze coronariche: tuttavia, esso richiede tempi di preparazione, di intervento e di guarigione piuttosto lunghi. In caso di infarto è necessario avere a disposizione una procedura più rapida che permetta di riaprire tempestivamente l'arteria ostruita e ristabilire l'adeguato flusso sanguigno, e si ricorre solitamente all'angioplastica coronarica: si tratta di una procedura di rivascularizzazione non chirurgica, che permette di rimuovere l'occlusione. Un palloncino viene immesso nel sistema cardiovascolare tramite l'arteria femorale o l'arteria radiale, e successivamente gonfiato quando raggiunge l'arteria coronaria occlusa, in modo da rimuovere la stenosi. Sulla superficie del palloncino viene solitamente posizionato uno *stent*, cioè una struttura metallica cilindrica a maglie, che consente di mantenere nel tempo la dilatazione ottenuta con il palloncino; possono essere utilizzati anche particolari tipi di *stent*, detti *stent medicati* (*drug eluting stents*), che al momento dell'apertura rilasciano un farmaco per contrastare la ristrenosi della coronaria.

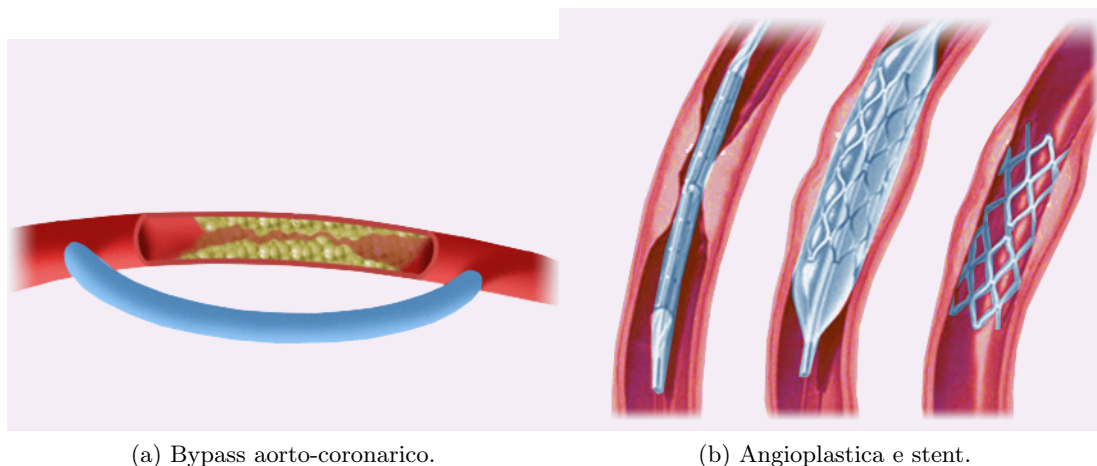
## 1.2 Modellistica matematica per l'emodinamica

Numerosi progressi sono stati compiuti negli scorsi decenni nel campo della modellistica matematica del sistema cardiovascolare, anche grazie alle crescenti risorse computazionali, a tecniche di ricostruzione delle immagini sempre più affidabili e a metodi numerici sempre più efficienti. Infatti, accanto alle tecniche sperimentali *in vitro* ed *in vivo*, tra cui laser Doppler (LDA), risonanza magnetica (MRI) e tomografia computerizzata (TC), si è sviluppato un ambito di ricerca in cui i flussi sanguigni vengono simulati numericamente grazie ai metodi della fluidodinamica computazionale. Questo approccio è particolarmente vantaggioso grazie alla minima invasività ed alla possibilità di ottenere indicazioni più accurate rispetto ai metodi sperimentali. Ad esempio, per quanto riguarda l'analisi del flusso nelle coronarie, i metodi sperimentali mostrano alcune criticità: è difficile infatti eseguire la misurazione delle quantità fisiche, sia a causa delle piccole dimensioni

del diametro delle arterie, sia a causa del movimento a cui le coronarie sono sottoposte durante il battito cardiaco.

Alcuni dei principali aspetti da tenere in considerazione nella descrizione matematica del sistema cardiovascolare, descritti in dettaglio in [FQV09, TD04, QTV00], sono ad esempio i seguenti:

1. *Interfaccia con i metodi sperimentali:* è necessario infatti effettuare una estrazione dei dati clinici, in particolare della geometria dei condotti vascolari, per ottenere risultati significativi; idealmente, questa operazione andrebbe fatta per ciascun paziente.
2. *Definizione di un modello matematico del sistema cardiovascolare:* il sistema cardiovascolare è notevolmente complesso (basti pensare alle differenze tra le dimensioni dei vasi che lo compongono, con variazioni di 4 ordini di grandezza tra il raggio dell'aorta e il raggio dei capillari). La sua modellizzazione matematica deve tenere conto dei seguenti aspetti:
  - *pulsatilità:* la soluzione di un modello stazionario spesso è troppo semplificata e poco rilevante per l'applicazione medica perché il flusso sanguigno è pulsatile; in prima approssimazione si può considerare il flusso come periodico, anche se questo è rigorosamente vero solo per brevi intervalli di tempo. Infatti considerando intervalli di tempo più ampi potrebbero insorgere componenti di non-periodicità, come ad esempio il passaggio da uno stato di quiete ad uno stato di sforzo fisico, a cui il cuore deve rispondere con una maggiore quantità di sangue in circolo. Sotto questa approssimazione, un ciclo cardiaco può essere diviso in due fasi: la sistole, che corrisponde all'intervallo di tempo in cui il cuore pompa il sangue nelle arterie, e la diastole, che corrisponde all'intervallo di tempo in cui il cuore riceve il sangue dalle vene.
  - *interazione fluido-struttura:* il flusso sanguigno interagisce meccanicamente con le pareti del vaso, in particolar modo nelle arterie di grosse dimensioni: è necessario dunque includere nel modello effetti di interazione fluido-struttura e fornire opportune leggi che governino la deformazione della parete. La natura pulsatile del flusso ha un ruolo anche nell'interazione fluido-struttura: infatti la parete si deforma durante la sistole, trasformando parte dell'energia meccanica del fluido in energia elastica, che viene restituita al flusso nella fase diastolica.
  - *condizioni al contorno:* solitamente si è interessati a studiare solo una parte del sistema cardiovascolare per mezzo di un modello fluidodinamico descritto da equazioni a derivate parziali tridimensionali, e dunque si necessita di opportune condizioni al contorno nell'interfaccia che collega il tratto in esame al resto del sistema circolatorio. Si ricorre solitamente a condizioni di Dirichlet (velocità imposta) sulla frontiera di inlet e condizioni di Neumann (sforzo normale imposto) sulla frontiera di outlet; entrambe queste scelte, pur garantendo sotto opportune ipotesi la buona posizione del modello matematico, presentano significativi problemi pratici: da un lato è infatti solitamente difficile misurare sperimentalmente i profili di velocità da imporre sulla frontiera di inlet, dall'altro è ancora dibattuta la scelta di quali siano le condizioni di outflow più adeguate. In alternativa, è possibile considerare l'intero sistema cardiovascolare, mantenendo un modello accurato (equazioni a derivate parziali definite su un dominio tridimensionale) per il tratto in esame ed utilizzando modelli ridotti (descritti, ad esempio, per mezzo di equazioni differenziali ordinarie, basati su un'analogia elettrica) per il resto del sistema cardiovascolare.
  - *reologia:* il sangue è un fluido molto complesso, formato da *elementi corpuscolati*, in particolare globuli rossi, globuli bianchi e piastrine, sospesi in liquido acquoso (*plasma*); inoltre, le proprietà del flusso sanguigno dipendono dalle dimensioni del condotto vascolare: se il flusso nelle arterie è solitamente modellato considerando il sangue come un fluido Newtoniano, nelle arterie più piccole (come ad esempio le coronarie) il flusso



(a) Bypass aorto-coronarico.

(b) Angioplastica e stent.

Figura 1.4: Modelli semplificati.

si può considerare laminare ed il numero di Reynolds è tale da rendere trascurabile la parte convettiva, ottenendo un comportamento governato dalle equazioni di Stokes. Tuttavia, in presenza di vasi di dimensioni più piccole, ed in particolare nei capillari, è necessario abbandonare il modello Newtoniano e considerare modelli reologici più complessi, come ad esempio un modello di fluido Newtoniano generalizzato (in cui la densità non è costante ma dipende dallo *shear rate*) o modelli viscoelastici.

3. *Sviluppo di metodi computazionali efficienti:* anche per quanto riguarda la simulazione numerica, i problemi di natura emodinamica pongono serie questioni legate agli oneri computazionali, in termini sia di tempo di calcolo, sia di capacità di memoria. In particolare:
  - la complessità della geometria richiede *mesh* molto fini;
  - la elevata variabilità temporale richiede adeguati metodi di avanzamento temporale;
  - la non linearità del modello matematico richiede la disponibilità di metodi adeguati per la risoluzione del problema discretizzato.

Le difficoltà di natura computazionale sono ancora più accentuate nel caso in cui si intenda risolvere un problema di controllo ottimo o di ottimizzazione di forma. In questi casi si ricorre infatti a metodi di tipo iterativo, in cui la soluzione del problema di stato ed il calcolo delle quantità di interesse vengono effettuati in modo ripetuto, comportando un notevole aumento dei tempi di calcolo. In particolare, nei problemi di ottimizzazione di forma anche la geometria (e conseguentemente la griglia di calcolo) viene aggiornata nel corso delle iterazioni, comportando la necessità di assemblare nuove strutture algebriche ad ogni passo.

### 1.3 Ottimizzazione di forma di un bypass aorto-coronarico

Questo lavoro di Tesi si propone di analizzare, mediante l'applicazione dei metodi dell'ottimizzazione di forma, il problema del *design* ottimale di un *graft* per un bypass aorto-coronarico. Questo studio necessita di una descrizione accurata del flusso all'interno del condotto vascolare e dell'innesto, eventualmente limitandosi a geometrie semplificate come quella mostrata in Figura 1.4(a): numerosi progressi sono stati compiuti verso una conoscenza più approfondita dell'emodinamica in corrispondenza degli innesti di un bypass al variare della geometria (angoli di incidenza, distanza dall'occlusione, ...), motivati dal fatto che una precisa analisi fluidodinamica sia indispensabile allo

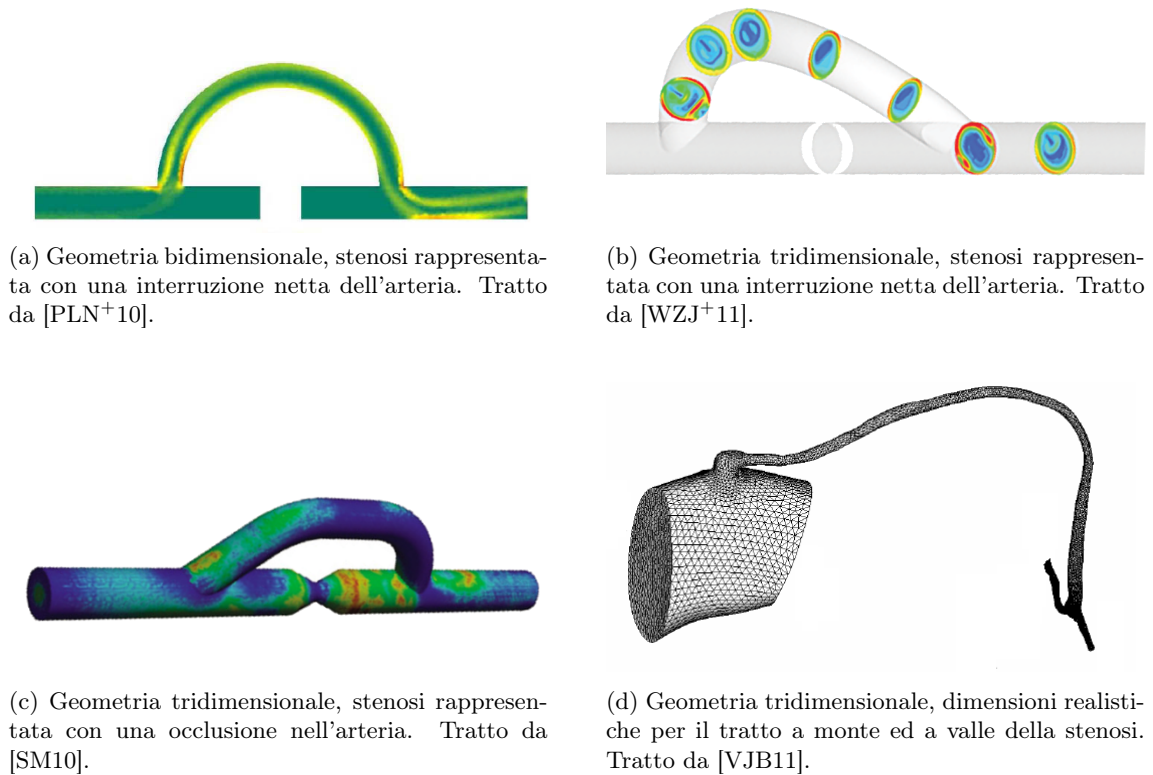


Figura 1.5: Alcune possibili geometrie semplificate di un bypass aorto-coronarico.

scopo di migliorare le prestazioni nel lungo periodo di questi dispositivi biomedicali. Si vedano, ad esempio, le analisi riportate in [DOYM11, WZJ<sup>+</sup>11, BDFD01, DBB01, PSC<sup>+</sup>08, BD00, LFB08, SCGT05]<sup>2</sup>.

Caratterizzando gli aspetti descritti nella sezione precedente al caso del flusso in un bypass aorto-coronarico, si possono ad esempio citare i seguenti elementi:

1. *Interfaccia con i metodi sperimentali:* numerose semplificazioni della geometria dei condotti vascolari sono state utilizzate per modellare il flusso sanguigno in un bypass: in ordine crescente di complessità, si può considerare un modello bidimensionale costituito dalla sezione longitudinale della coronaria e del *graft* (Figura 1.5(a), [PLN<sup>+</sup>10]). Una geometria più accurata è quella di Figura 1.5(b) (si veda [WZJ<sup>+</sup>11]), in cui si considera il problema tridimensionale, o quella di Figura 1.5(c) (tratta da [SM10]) in cui la stenosi non è rappresentata come una interruzione netta della coronaria, ma come una parziale occlusione che restringe il diametro vascolare. Una geometria ancora più realistica è quella impiegata in [VJB11], e mostrata in Figura 1.5(d), in cui si tiene conto che il *graft* viene suturato all'aorta, che ha dimensioni notevolmente diverse rispetto alle arterie coronarie. In questa Tesi verranno considerate geometrie tridimensionali simili a quelle di Figura 1.5(c).
2. *Definizione di un modello matematico per il flusso nelle coronarie e nel graft:*
  - o *pulsatilità:* sono stati condotti studi sia sul modello stazionario (si vedano, ad esempio, [BD00, FTR94, HCHH96, IMP96]), sia sul modello non stazionario (si vedano, ad esempio, [LAK97, DOYM11, BDFD01, SM10, PSC<sup>+</sup>08, LFB08]), più completo in quanto permette di ottenere una descrizione accurata del flusso nelle fasi di sistole e

<sup>2</sup>Simili progressi sono stati compiuti anche nel contesto dell'angioplastica coronarica con le geometrie semplificate di Figura 1.4(b): si vedano, ad esempio, le analisi riportate in [DZ08, ZDP<sup>+</sup>09, MGP<sup>+</sup>07].

diastole: per questo motivo, anche in questa Tesi, le simulazioni numeriche saranno basate su un modello non stazionario; per semplicità, e per validare l'implementazione del risolutore su un problema più semplice, è stata predisposta anche una prima fase di simulazioni sul modello stazionario.

- *interazione fluido-struttura*: a causa delle piccole dimensioni delle coronarie e del *graft*, in questa Tesi appare ragionevole trascurare l'interazione fluido-struttura; infatti, l'elasticità delle pareti è particolarmente rilevante solo nelle arterie di grosse dimensioni, come l'aorta (si veda [FQV09]). Questa ipotesi è condivisa anche con gli studi citati in precedenza. Appare invece meno trascurabile un altro tipo di interazione, data dal movimento a cui l'epicardio (e dunque anche le coronarie) è soggetto a causa del battito cardiaco: tuttavia, prima di poter considerare questa interazione, sarebbe necessario avere a disposizione una geometria notevolmente più realistica di quella di Figura 1.5(c) (a parte le differenti dimensioni dei vasi, andrebbe ad esempio considerato che la superficie dell'epicardio è curva, e quindi considerare una curvatura nella coronaria occlusa).
- *condizioni al contorno*: la maggiore criticità è la scelta di un profilo di velocità adeguata nella sezione di inlet. Alcuni studi (ad esempio [BDFD01]) presentano una prima fase di raccolta di dati sperimentali ed una seconda fase di simulazione mediante l'emodinamica computazionale. Anche in questa Tesi si è fatto uso, nelle simulazioni non stazionarie, di profili di velocità realistici nella sezione di inlet, tratti da lavori precedenti.
- *reologia*: uno studio di modelli non-Newtoniani è stato ad esempio effettuato in [ABH05]; tuttavia, come riportato nella sezione precedente, il sangue nelle arterie può essere considerato un fluido Newtoniano, e dunque è solitamente lecito utilizzare il modello fluidodinamico descritto dalle equazioni di Navier-Stokes (come riportato, ad esempio, in [LAK97, DOYM11, BDFD01, PLN<sup>+</sup>10, SM10, PSC<sup>+</sup>08, LFB08]). In questa Tesi verrà effettuata l'ottimizzazione della forma del bypass supponendo che il flusso sanguigno sia governato, invece, dal modello lineare fornito dal problema di Stokes: questa ipotesi, comunque giustificata dalle piccole dimensioni delle coronarie, è stata dettata dalla necessità di contenere l'onere computazionale delle simulazioni; tuttavia verrà effettuata una ulteriore fase di validazione risolvendo un problema di Navier-Stokes sulla forma ottima ottenuta con il modello di Stokes, confrontando i risultati ottenuti con gli studi citati in precedenza.

### 3. Sviluppo di metodi computazionali efficienti:

- le simulazioni sono state effettuate mediante la libreria *LifeV*, che fornisce tutti gli strumenti necessari per ottenere una risoluzione efficiente dei modelli fluidodinamici; il lavoro di questa Tesi ha anche permesso di aggiungere alla libreria le funzionalità per la risoluzione di un generico problema di ottimizzazione di forma;
- la complessità della geometria e del profilo temporale di inlet richiedono *mesh* e passi temporali fini; in questa Tesi questo vincolo è stato rispettato, compatibilmente con le risorse computazionali a disposizione, con particolare attenzione alla fase di validazione con modello di Navier-Stokes.

Il problema del *design* di un *graft* aorto-coronarico si presta particolarmente all'applicazione delle tecniche dell'ottimizzazione di forma: la soluzione di un problema di questo tipo può fornire utili indicazioni alla comunità medica dal momento che è stato mostrato che le caratteristiche geometriche dell'innesto hanno infatti una forte influenza sull'emodinamica nella regione del bypass, ed è ormai consolidato il legame presente tra alcune quantità di natura fluidodinamica ed il processo di ristenosi. Agli aspetti della modellazione del fenomeno emodinamico, citati in precedenza, si aggiungono dunque altre considerazioni per stabilire come eseguire il confronto tra



due configurazioni geometriche diverse ed affermare, a partire da un opportuno *funzionale costo* dipendente dalle grandezze fluidodinamiche, quale delle due sia “clinicamente migliore”. Numerosi studi (si veda ad esempio [Roz05b, LFB08]) hanno permesso di evidenziare il legame presente tra i seguenti *indici* fluidodinamici ed il processo di ristenosi:

- *regolarità dei profili di velocità*: i risultati riportati in [DOYM11] mostrano il legame tra la bassa regolarità del flusso – in particolare la presenza di zone di separazione, ricircolo e ristagno in prossimità dei punti di sutura – ed il processo di ristenosi. Per questo motivo si può considerare un funzionale costo basato sulla vorticità  $\nabla \times \mathbf{u}$  (dove  $\mathbf{u}$  indica la velocità del fluido) e, seguendo [Roz05b, QR03, Roz05a, AQR07], minimizzare il modulo di questa quantità sull’intero dominio o sulla zona a valle dell’occlusione.
- *sforzi di taglio a parete*: il processo di ristenosi è influenzato anche dalle forze agenti sulle pareti. Un indice di particolare rilevanza è lo sforzo di taglio a parete (*wall shear stress*, WSS), definito come

$$\mathbf{WSS} = \mathbf{t} - (\mathbf{t} \cdot \mathbf{n})\mathbf{n}, \quad \mathbf{t} = T\mathbf{n},$$

dove  $T$  è il tensore degli sforzi del modello fluidodinamico e  $\mathbf{t}$  è la forza esercitata sulla parete di normale  $\mathbf{n}$ . Gli studi citati in [LFB08] hanno infatti mostrato che le zone in cui lo sforzo di taglio a parete è basso sono maggiormente soggette al processo di ristenosi; in tal proposito le regioni più critiche sono quelle poste sulla superficie immediatamente a valle dell’anastomosi distale. Si potrebbe dunque scegliere di massimizzare un opportuno valore medio del WSS: tuttavia, questa scelta appare clinicamente rischiosa, perché è stato anche osservato che la presenza di un elevato sforzo di taglio a parete è indice della formazione di trombi. È dunque necessario considerare l’ottimizzazione di un funzionale costo più complesso: ad esempio, seguendo [MFT08, SM10], si può minimizzare la misura della superficie in cui il WSS è inferiore ad una fissata soglia critica, al di sopra della quale il processo di ristenosi agisce a livelli accettabili.

- *oscillating shear index*: per un flusso non stazionario di periodo  $T$ , si definisce *oscillating shear index* (OSI) il seguente indice, introdotto originariamente in [KGZG85]:

$$OSI = \frac{1}{2} \left[ 1 - \frac{\left| \int_0^T \mathbf{WSS}(t) dt \right|}{\int_0^T |\mathbf{WSS}(t)| dt} \right], \quad 0 \leq OSI \leq 0.5.$$

Il valore nullo corrisponde a trazione (e quindi flusso) unidirezionale; il valore massimo corrisponde a trazione media nulla, tale per cui la direzione del vettore  $\mathbf{WSS}$  cambia frequentemente: è stato mostrato (si vedano i riferimenti contenuti in [Ku97]) che queste oscillazioni contribuiscono al processo di ristenosi. In conclusione, una possibile scelta per il problema di ottimizzazione può essere quella di minimizzare una opportuna media superficiale dell’OSI.

In questa Tesi si farà riferimento ad un indice che garantisca una migliore regolarità dei profili di velocità, mediante il funzionale costo dato dall’energia dissipata (dipendente da  $\nabla \mathbf{u}$ ): questa scelta – condivisa con [PLN<sup>+</sup>10] – è anche parzialmente giustificata dal fatto che, sotto opportune ipotesi semplificative che verranno precisate nella Sezione 3.1, si può mostrare che minimizzare l’energia dissipata equivale a minimizzare la vorticità del flusso. Gli indici WSS ed OSI verranno inoltre valutati nella fase di validazione con equazioni di Navier-Stokes, confrontando i risultati ottenuti con le necessità biomediche di ridurre, per quanto possibile, il processo di ristenosi.

## Capitolo 2

# Formulazione astratta di un problema di ottimizzazione di forma

Dopo un breve richiamo alla teoria dei problemi di controllo ottimo per equazioni a derivate parziali (Sezione 2.1), viene presentata in questo capitolo la formulazione astratta di un problema di ottimizzazione di forma (visti come caso particolare di problemi di controllo ottimo), l'analisi di esistenza ed un metodo per caratterizzare la forma ottima mediante opportune *condizioni di ottimalità*.

In particolare:

1. nella Sezione 2.2 viene enunciato un risultato astratto di esistenza (Teorema 2.4), le cui ipotesi ed assunzioni sono illustrate nelle Sezioni da 2.2.1 a 2.2.3, mostrandone infine nella Sezione 2.2.4 un'applicazione ad un problema modello.
2. nella Sezione 2.3 viene introdotto un metodo generale per la descrizione dei domini, basato su *perturbazioni dell'identità* di un dominio di riferimento fissato. Questo metodo permette di definire in modo naturale la nozione di derivata rispetto alla forma (Sezione 2.3.2), necessaria per la scrittura delle condizioni di ottimalità, alle quali verrà dedicata la Sezione 2.4. Un caso particolare di perturbazione dell'identità, conveniente specialmente da un punto di vista computazionale, è la mappa *Free Form Deformation* che verrà illustrata nella Sezione 2.3.1 e successivamente impiegata nelle simulazioni numeriche presentate nel Capitolo 5. In particolare, il Teorema 2.4 di esistenza verrà applicato alla classe dei domini ottenuti deformando il dominio di riferimento tramite la mappa *Free Form Deformation*.

## 2.1 Problemi di controllo ottimo per EDP

Come anticipato nel capitolo introduttivo, l'obiettivo di un problema di controllo ottimo per EDP è quello di agire su un sistema, descritto da un problema differenziale alle derivate parziali, in modo da ottenere un comportamento desiderato. Da un punto di vista astratto, un problema di controllo ottimo per EDP può essere scritto come la minimizzazione di un funzionale costo  $J(\cdot)$ , dipendente dalla soluzione di una EDP, su un opportuno insieme di controlli ammissibili  $\mathcal{U}_{ad}$ : determinare, se esiste, un controllo  $\hat{u} \in \mathcal{U}_{ad}$  tale che

$$\hat{u} = \arg \min_{u \in \mathcal{U}_{ad}} J(u).$$

In particolare:

1. la **variabile di controllo**  $u$  appartiene all'insieme dei controlli ammissibili  $\mathcal{U}_{ad}$ . Tipicamente  $\mathcal{U}_{ad}$  è un sottoinsieme convesso e chiuso di un opportuno spazio di Banach riflessivo  $\mathcal{U}$ ;



2. la **variabile di stato**  $y = y(u)$ , dipendente dal controllo  $u$ , appartiene ad uno spazio di Hilbert  $\mathcal{V}$  e soddisfa la seguente equazione (**problema di stato**):

$$A(u)y(u) = f + Bu;$$

$A : \mathcal{V} \rightarrow \mathcal{V}'$  è un'operatore differenziale (corredato da condizioni al contorno e, nel caso non stazionario, condizioni iniziali; solitamente si tratta di un operatore ellittico del secondo ordine) eventualmente dipendente da  $u$  (ad esempio tramite un coefficiente), dove  $\mathcal{V}'$  indica lo spazio duale  $\mathcal{V}$ ,  $f \in \mathcal{V}'$  e  $B : \mathcal{U} \rightarrow \mathcal{V}'$  è detta *funzione di controllo*;

3. la **funzione di osservazione**  $z(u) = Cy(u)$ , con  $C : \mathcal{V} \rightarrow \mathcal{Z}$  operatore di osservazione e  $\mathcal{Z}$  opportuno spazio di Hilbert (spazio delle *funzioni osservate*);
4. il **funzionale costo**, che rappresenta l'obiettivo da ottimizzare, dipende dal controllo  $u$  sia esplicitamente, sia implicitamente tramite l'osservazione  $z$ . In generale esso è dato da

$$J(u) = F(z(u), u), \quad \forall u \in \mathcal{U}_{ad},$$

dove  $F : \mathcal{Z} \times \mathcal{U} \rightarrow \mathbb{R}$ .

Considerando ad esempio il seguente problema di stato:

$$\begin{cases} -\Delta y = f + u, & \text{in } \Omega, \\ y = 0, & \text{su } \partial\Omega, \end{cases} \quad (2.1)$$

con  $\Omega \subset \mathbb{R}^n$  aperto,  $y \in \mathcal{V} = H_0^1(\Omega)$  variabile di stato,  $f \in L^2(\Omega)$ ,  $u \in \mathcal{U}_{ad} = \mathcal{U} = L^2(\Omega)$  variabile di controllo, si può cercare di minimizzare il funzionale

$$J(u) = \int_{\Omega} (y(u) - z_d)^2 dx,$$

essendo  $z_d \in L^2(\Omega)$  il “valore desiderato” per lo stato del sistema. Si parla in questo caso di problema di *controllo distribuito con osservazione distribuita*. È possibile inoltre descrivere anche problemi di *controllo al bordo*, in cui la variabile di controllo  $u \in \mathcal{U}_{ad} = \mathcal{U} = H^{1/2}(\partial\Omega)$  è definita sul bordo  $\partial\Omega$  ed influenza le condizioni al contorno del problema di stato

$$\begin{cases} -\Delta y = f, & \text{in } \Omega, \\ y = u, & \text{su } \partial\Omega, \end{cases}$$

oppure problemi con *osservazione al bordo*, in cui si è interessati alla minimizzazione di un funzionale costo della forma

$$J(u) = \int_{\partial\Omega} (y(u) - z_d)^2 ds,$$

dove  $y(u)$  è soluzione del problema di stato (2.1) e  $z_d \in H^{1/2}(\partial\Omega)$ .

### 2.1.1 Alcuni risultati per la minimizzazione di funzionali

In questa sezione vengono ripresi alcuni risultati relativi alla minimizzazione di funzionali quadratici in spazi di Hilbert. Si consideri dunque uno spazio di Hilbert  $\mathcal{U}$ , su cui è definita una forma bilineare  $\pi(\cdot, \cdot) : \mathcal{U} \times \mathcal{U} \rightarrow \mathbb{R}$ , che goda delle seguenti proprietà:

- o simmetria, cioè

$$\pi(u, v) = \pi(v, u), \quad \forall u, v \in \mathcal{U};$$

o continuità, cioè esiste una costante  $M > 0$  tale che

$$|\pi(u, v)| \leq M \|u\|_{\mathcal{U}} \|v\|_{\mathcal{U}}, \quad \forall u, v \in \mathcal{U};$$

o coercività, cioè esiste una costante  $\alpha > 0$  tale che

$$\pi(v, v) \geq \alpha \|v\|_{\mathcal{U}}^2, \quad \forall v \in \mathcal{U},$$

ed un funzionale  $F : \mathcal{U} \rightarrow \mathbb{R}$  lineare e continuo.

Dato un insieme di controlli ammissibili  $\mathcal{U}_{ad} \subset \mathcal{U}$ , convesso e chiuso, si consideri un funzionale costo  $J$  della forma seguente:

$$J(v) = \pi(v, v) - 2F(v), \quad \forall v \in \mathcal{U}_{ad}. \quad (2.2)$$

Vale allora il seguente risultato di esistenza:

**Teorema 2.1 ([Lio71]).** *Siano  $\pi(u, v)$  forma bilineare continua, coerciva e simmetrica,  $F$  funzionale lineare e continuo, e  $\mathcal{U}_{ad}$  sottoinsieme convesso e chiuso dello spazio di Hilbert  $\mathcal{U}$ . Allora esiste un unico controllo  $\hat{u} \in \mathcal{U}_{ad}$  tale che*

$$\hat{u} = \arg \min_{u \in \mathcal{U}_{ad}} J(u),$$

dove il funzionale costo  $J$  è dato da (2.2).

Per la dimostrazione di questo risultato si rimanda a [Lio71]; in particolare l'esistenza di un controllo ottimo si basa su una proprietà di **compattezza di  $\mathcal{U}_{ad}$**  (che discende dalle ipotesi di convessità e chiusura in  $\mathcal{U}$ ) e sulla **semicontinuità inferiore di  $J$**  (che discende dalla continuità di  $\pi$  ed  $F$ ). Queste due proprietà saranno alla base del Teorema 2.4 di esistenza per problemi di ottimizzazione di forma, che verrà introdotto nella Sezione 2.2.

Dopo aver mostrato l'esistenza (e l'unicità) di un controllo ottimo, è necessario caratterizzarlo mediante opportune *condizioni di ottimalità*, come riportato nel seguente risultato:

**Teorema 2.2 ([Lio71]).** *Siano  $\pi(u, v)$  forma bilineare continua, coerciva e simmetrica,  $F$  funzionale lineare e continuo, e  $\mathcal{U}$  spazio di Hilbert:*

(i) *se  $\mathcal{U}_{ad}$  è un sottoinsieme convesso e chiuso di  $\mathcal{U}$ , allora il controllo ottimo  $\hat{u} \in \mathcal{U}_{ad}$  soddisfa la seguente disequazione variazionale*

$$\pi(\hat{u}, v - \hat{u}) \geq F(v - \hat{u}), \quad \forall v \in \mathcal{U}_{ad}; \quad (2.3)$$

(ii) *se  $\mathcal{U}_{ad}$  è un cono convesso e chiuso di  $\mathcal{U}$ , con vertice nell'origine, allora il controllo ottimo  $\hat{u} \in \mathcal{U}_{ad}$  soddisfa*

$$\pi(\hat{u}, v) \geq F(v), \quad \forall v \in \mathcal{U}_{ad}, \quad e \quad \pi(\hat{u}, \hat{u}) = F(\hat{u});$$

(iii) *se  $\mathcal{U}_{ad} = \mathcal{U}$ , allora il controllo ottimo  $\hat{u} \in \mathcal{U}_{ad}$  soddisfa la seguente equazione di Eulero*

$$\pi(\hat{u}, v) = F(v), \quad \forall v \in \mathcal{U}.$$

Questi risultati verranno applicati nelle Sezioni 2.1.2 e 2.1.3 per caratterizzare il controllo ottimo mediante due differenti approcci, alla Lions e mediante moltiplicatori di Lagrange, rispettivamente.

### 2.1.2 Controllo ottimo per problemi ellittici: approccio alla Lions

In questa sezione vengono riportati alcuni risultati relativi al controllo ottimo per problemi ellittici, seguendo il cosiddetto *approccio alla Lions* [Lio71].

Siano  $\mathcal{V}$  ed  $\mathcal{H}$  due spazi di Hilbert, tali che  $\mathcal{V}$  risulti denso ed immerso con continuità in  $\mathcal{H}$ , e siano  $\mathcal{V}'$  ed  $\mathcal{H}'$  i rispettivi spazi duali. Indicando con  $a(\cdot, \cdot) : \mathcal{V} \times \mathcal{V} \rightarrow \mathbb{R}$  una forma bilineare continua e coerciva, e con  $L \in \mathcal{V}'$  un funzionale lineare e continuo,  $L(\psi) = (f, \psi)_{\mathcal{H}}$  per qualche  $f \in \mathcal{H}$ , il Lemma di Lax-Milgram assicura che esiste una unica soluzione  $y \in \mathcal{V}$  del problema

$$a(y, \psi) = (f, \psi)_{\mathcal{H}}, \quad \forall \psi \in \mathcal{V};$$

in modo equivalente, questo problema può anche essere espresso in forma operatoriale come

$$Ay = f, \quad \text{in } \mathcal{V}',$$

dove  $A$  è l'operatore lineare e continuo tale che

$$\langle A\phi, \psi \rangle = a(\phi, \psi), \quad \forall \phi, \psi \in \mathcal{V}',$$

Il problema di stato può dunque essere espresso in forma operatoriale come

$$Ay = f + Bu, \quad \text{in } \mathcal{V}',$$

dove  $B : \mathcal{U} \rightarrow \mathcal{V}'$  è la funzione di controllo. Introducendo la funzione di osservazione  $C : \mathcal{V} \rightarrow \mathcal{Z}$ , con  $\mathcal{Z}$  spazio di Hilbert, è possibile associare ad ogni controllo  $u$  il seguente funzionale costo:

$$J(u) = J(y(u), u) = \|Cy(u) - z_d\|_{\mathcal{Z}}^2 + (Nu, u)_{\mathcal{U}},$$

in cui  $z_d$  è rappresenta una condizione desiderata ed il secondo termine costituisce una penalità sul controllo, dove  $N : \mathcal{U} \rightarrow \mathcal{U}$  è una forma definita positiva, cioè tale che  $(Nu, u)_{\mathcal{U}} \geq \nu \|u\|_{\mathcal{U}}^2$  per qualche  $\nu > 0$ .

Al fine di applicare i risultati presentati nella Sezione 2.1.1, si noti che è possibile riscrivere il funzionale costo  $J$  nel seguente modo

$$J(u) = J(y(u), u) = \|C[y(u) - y(0)] + Cy(0) - z_d\|_{\mathcal{Z}}^2 + (Nu, u)_{\mathcal{U}},$$

e definire  $\pi(u, v)$  e  $F(v)$ , rispettivamente, come

$$\begin{aligned} \pi(u, v) &= (C[y(u) - y(0)], C[y(v) - y(0)])_{\mathcal{Z}} + (Nu, v)_{\mathcal{U}}, \\ F(v) &= -(C[y(v) - y(0)], Cy(0) - z_d)_{\mathcal{Z}}. \end{aligned}$$

Ciò permette di esprimere il funzionale costo, a meno di una costante positiva (ininfluente per l'ottimizzazione), come

$$J(u) = \pi(u, v) - 2F(v).$$

In particolare

- o la bilinearità di  $\pi$  e la linearità di  $F$  sono conseguenza del fatto che l'operatore  $A$  è un isomorfismo continuo da  $\mathcal{V}$  a  $\mathcal{V}'$  (grazie alla buona posizione del problema di stato) e che l'applicazione

$$v \mapsto y(v) - y(0) = A^{-1}B(v)$$

è lineare;

o la coercività di  $\pi(u, v)$  è garantita dalla definita positività del termine di penalità.

È quindi possibile applicare il Teorema 2.1 per dimostrare l'esistenza di un unico controllo ottimo  $\hat{u}$ . Le condizioni di ottimalità si ricavano invece dal Teorema 2.2: infatti, sostituendo le espressioni di  $\pi(u, v)$  e  $F(v)$  nella (2.3), si ottiene

$$(Cy(\hat{u}) - z_d, C[y(v) - y(u)])_{\mathcal{Z}} + (N\hat{u}, v - \hat{u})_{\mathcal{U}} \geq 0, \quad \forall v \in \mathcal{U}_{ad}.$$

Sia ora  $C' : \mathcal{Z}' \rightarrow \mathcal{V}'$  l'operatore aggiunto di  $C$ , cioè

$$\langle C'\phi, v \rangle_{\mathcal{V}' \times \mathcal{V}} = \langle \phi, Cv \rangle_{\mathcal{Z}' \times \mathcal{Z}}, \quad \forall v \in \mathcal{V}, \phi \in \mathcal{Z}';$$

introducendo l'isomorfismo canonico  $\Lambda_{\mathcal{Z}} : \mathcal{Z} \rightarrow \mathcal{Z}'$ , la condizione di ottimalità si può riscrivere come

$$\langle C'\Lambda_{\mathcal{Z}}[Cy(\hat{u}) - z_d], y(v) - y(\hat{u}) \rangle_{\mathcal{V}' \times \mathcal{V}} + (N\hat{u}, v - \hat{u})_{\mathcal{U}} \geq 0, \quad \forall v \in \mathcal{U}_{ad}. \quad (2.4)$$

Sia ora  $A' : \mathcal{V}' \rightarrow \mathcal{V}$  l'operatore aggiunto di  $A$ : definendo lo *stato aggiunto*  $p = p(v)$  soluzione del *problema aggiunto*

$$A'p(v) = C'\Lambda_{\mathcal{Z}}[Cy(v) - z_d] \quad \text{in } \mathcal{V}',$$

e sostituendo questa espressione nella condizione di ottimalità (2.4) si ottiene, dopo qualche passaggio,

$$\langle B'p(\hat{u}), v - \hat{u} \rangle_{\mathcal{U}' \times \mathcal{U}} + (N\hat{u}, v - \hat{u})_{\mathcal{U}} \geq 0, \quad \forall v \in \mathcal{U}_{ad},$$

dove  $B' : \mathcal{V} \rightarrow \mathcal{U}'$  indica l'operatore aggiunto della funzione di controllo.

Si ottiene infine, essendo  $\Lambda_{\mathcal{U}}$  l'isomorfismo canonico di  $\mathcal{U}$  in  $\mathcal{U}'$ , la seguente condizione:

$$(\Lambda_{\mathcal{U}}^{-1}B'p(\hat{u}) + N\hat{u}, v - \hat{u})_{\mathcal{U}} \geq 0, \quad \forall v \in \mathcal{U}_{ad};$$

essa fornisce in particolare l'espressione del gradiente di  $J$ , definito come l'elemento di Riesz che rappresenta  $J'$ :

$$\nabla J(\hat{u}) = \Lambda_{\mathcal{U}}^{-1}B'p(\hat{u}) + N\hat{u}.$$

La valutazione del gradiente del funzionale costo, necessaria per la costruzione di uno schema numerico per l'approssimazione del controllo ottimo  $\hat{u}$ , può essere dunque effettuata risolvendo un ulteriore problema alle derivate parziali.

### 2.1.3 Controllo ottimo per problemi ellittici: il metodo dei moltiplicatori di Lagrange

Interpretando il problema di controllo ottimo per EDP come un problema di ottimizzazione vincolata, in cui il vincolo è dato dal problema di stato, è possibile usare il metodo dei moltiplicatori di Lagrange per ottenere le condizioni di ottimalità in modo automatico, differenziando un opportuno funzionale Lagrangiano, come nel caso dell'ottimizzazione vincolata di funzioni di  $n$  variabili reali. In questa sezione viene introdotto il metodo dei moltiplicatori di Lagrange da un punto di vista astratto, e successivamente applicato ad un problema di controllo ottimo per EDP ellittiche.

Siano  $X, Y, \mathcal{U}$  spazi di Banach riflessivi,  $V \subset X$  aperto e  $\mathcal{U}_{ad} \subset \mathcal{U}$ , e si consideri il seguente problema di ottimizzazione vincolata

$$\begin{aligned} & \min F(x, u) \\ & \text{s.t. } G(x, u) = 0, \quad x \in V, \quad u \in \mathcal{U}_{ad}, \end{aligned} \quad (2.5)$$

per il funzionale costo  $F : X \times \mathcal{U} \rightarrow \mathbb{R}$  e l'operatore  $G : X \times \mathcal{U} \rightarrow Y$ , che rappresenta il vincolo.

Se il funzionale  $F$  è della forma

$$F(x, u) = P(x) + N(u),$$

con  $P : X \rightarrow \mathbb{R}$  ed  $N : \mathcal{U} \rightarrow \mathbb{R}$  allora, sotto opportune condizioni, il problema di ottimizzazione vincolata (2.5) ammette un punto di minimo, come enunciato nel seguente Teorema:

**Teorema 2.3.** *Si assuma che:*

1.  $\inf_{x \in X} P = p_0 > -\infty$  ed esiste una coppia  $(x, u) \in V \times \mathcal{U}_{ad}$  tale che  $G(x, u) = 0$ ;
2. esistono  $\beta, \gamma > 0$  tali che  $N(u) \geq \beta \|u\|_{\mathcal{U}}^\gamma$  per ogni  $u \in \mathcal{U}_{ad}$ ;
3. date le successioni  $\{u_k\} \subset \mathcal{U}_{ad}$  e  $\{x_k\} \subset X$ , se  $u_k \xrightarrow{\mathcal{U}} u$  e  $x_k \xrightarrow{X} x$ , allora  $G(x_k, u_k) \xrightarrow{Y} G(x, u)$ ;
4.  $F(x, u) = P(x) + N(u)$  è sequenzialmente debolmente inferiormente semicontinuo in  $X \times \mathcal{U}_{ad}$ ;
5. date le successioni  $\{u_k\} \subset \mathcal{U}_{ad}$  e  $\{x_k\} \subset X$ , se  $|P(x_k)| \leq M$  e  $G(x_k, u_k) = 0$ , allora esiste  $M_0 > 0$  tale che  $\|x_k\|_X \leq M_0$ .

Allora esiste una soluzione  $(\hat{x}, \hat{u})$  del problema di ottimizzazione vincolata

$$\begin{aligned} & \min P(x) + N(u) \\ & \text{s.t. } G(x, u) = 0, \quad x \in V, \quad u \in \mathcal{U}_{ad}. \end{aligned}$$

Come per il Teorema 2.1, anche in questo caso la dimostrazione è basata su una proprietà di compattezza e sulla semicontinuità inferiore del funzionale obiettivo; tuttavia in questo caso le ipotesi non sono sufficienti a garantire l'unicità.

Per ricavare le condizioni di ottimalità è possibile introdurre il seguente funzionale Lagrangiano associato al problema di ottimizzazione non vincolata (2.5):

$$\mathcal{L}(x, u; p, \lambda_0) = \lambda_0 F(x, u) + \langle p, G(x, u) \rangle_*,$$

dove  $\langle \cdot, \cdot \rangle_*$  indica la dualità tra  $Y$  ed  $Y'$ ,  $p \in Y'$  è il moltiplicatore di Lagrange del vincolo  $G(x, u) = 0$  e  $\lambda_0 = 0$  oppure  $\lambda_0 = 1$ . Sotto opportune ipotesi di convessità e differenziabilità nel senso di Fréchet (si veda, ad esempio, [Gun03]), è possibile mostrare che esiste  $\hat{p} \in Y'$  tale che la coppia ottima  $(\hat{x}, \hat{u})$  ed il moltiplicatore ottimo  $\hat{p}$  soddisfano il seguente sistema:

$$\begin{cases} \mathcal{L}_x(\hat{x}, \hat{u}; \hat{p}, 1)x = F_x(\hat{x}, \hat{u})x + \langle \hat{p}, G_x(\hat{x}, \hat{u})x \rangle_* = 0, & \forall x \in X, \\ \mathcal{L}_p(\hat{x}, \hat{u}; \hat{p}, 1)p = \langle p, G(\hat{x}, \hat{u}) \rangle_* = 0, & \forall p \in Y', \\ \min_{u \in \mathcal{U}_{ad}} \mathcal{L}_p(\hat{x}, u; \hat{p}, 1) = \mathcal{L}_p(\hat{x}, \hat{u}; \hat{p}, 1), \end{cases} \quad (2.6)$$

dove  $\mathcal{L}_x$  indica la derivata di Gâteaux di  $\mathcal{L}$  rispetto ad  $x$ .

Introducendo l'operatore aggiunto  $G'_x(\hat{x}, \hat{u}) : Y' \rightarrow X'$ , ed evidenziando la dualità tra  $X$  ed  $X'$ , è possibile riscrivere la prima equazione del sistema (2.6) nella forma

$$\langle F_x(\hat{x}, \hat{u}) + G'_x(\hat{x}, \hat{u})\hat{p}, x \rangle_* = 0, \quad \forall x \in X,$$

da cui si ricava il seguente *problema aggiunto*:

$$F_x(\hat{x}, \hat{u}) + G'_x(\hat{x}, \hat{u})\hat{p} = 0.$$

La seconda equazione del sistema (2.6) equivale al vincolo  $G(\hat{x}, \hat{u}) = 0$ ; la terza esprime il principio di minimo. In particolare, se  $u \mapsto G(\cdot, u)$  è differenziabile secondo Fréchet, convesso e

debolmente semicontinuo inferiormente, il principio di minimo può essere scritto sotto forma di disuguaglianza variazionale per  $\hat{u}$ :

$$F_u(\hat{x}, \hat{u})(v - \hat{u}) + \langle \hat{p}, G_u(\hat{x}, \hat{u})(v - \hat{u}) \rangle_* \geq 0, \quad \forall v \in \mathcal{U}_{ad};$$

introducendo l'operatore aggiunto  $G'_u(\hat{x}, \hat{u}) : Y' \rightarrow \mathcal{U}'$  si può scrivere

$$\langle F_u(\hat{x}, \hat{u}) + G'_u(\hat{x}, \hat{u})\hat{p}, v - \hat{u} \rangle_{\mathcal{U}' \times \mathcal{U}} \geq 0, \quad \forall v \in \mathcal{U}_{ad}.$$

che si riduce ad un'equazione se  $\mathcal{U}_{ad} = \mathcal{U}$ .

Infine, se  $\mathcal{U}$  è uno spazio di Hilbert, introducendo l'isomorfismo canonico  $\Lambda_{\mathcal{U}} : \mathcal{U} \rightarrow \mathcal{U}'$ , si deduce la seguente formula

$$\nabla F_u(\hat{x}, \hat{u}) + \Lambda_{\mathcal{U}}^{-1} G'_u(\hat{x}, \hat{u})\hat{p} = 0,$$

dalla quale è possibile estrarre le informazioni necessarie per l'approssimazione numerica mediante una procedura iterativa di ottimizzazione basata sul gradiente del funzionale costo.

Il problema di controllo ottimo per EDP ellittiche descritto nella Sezione 2.1.2, può essere dunque visto come un problema di ottimizzazione vincolata:

$$\begin{aligned} \min J(y, u) &= \|Cy - z_d\|_{\mathcal{Z}}^2 + (Nu, u)_{\mathcal{U}} \\ \text{s.t. } Ay &= f + Bu, \quad \forall u \in \mathcal{U}_{ad}. \end{aligned}$$

Introducendo il funzionale Lagrangiano  $\mathcal{L} : \mathcal{V} \times \mathcal{U} \times \mathcal{V} \rightarrow \mathbb{R}$  dato da

$$\mathcal{L}(y, u; p) = J(y, u) + \langle p, f + Bu - Ay \rangle_*,$$

il sistema (2.6) non è altro che il sistema di Eulero-Lagrange per il funzionale Lagrangiano  $\mathcal{L}$ : la soluzione del problema di controllo ottimo può essere dunque caratterizzata dal seguente sistema di equazioni:

$$\begin{cases} \mathcal{L}_y(\hat{y}, \hat{u}; \hat{p})\phi = 0, & \forall \phi \in \mathcal{V}, & \text{problema aggiunto,} \\ \mathcal{L}_p(\hat{y}, \hat{u}; \hat{p})\varphi = 0, & \forall \varphi \in \mathcal{V}, & \text{problema di stato,} \\ \mathcal{L}_u(\hat{y}, \hat{u}; \hat{p})\psi = 0, & \forall \psi \in \mathcal{U}_{ad}, & \text{equazione di sensitività.} \end{cases} \quad (2.7)$$

In modo analogo, indicando con  $a(\cdot, \cdot)$ ,  $b(\cdot, \cdot)$  ed  $n(\cdot, \cdot)$  le forme bilineari associate agli operatori  $A$ ,  $B$  ed  $N$  rispettivamente, e considerando il funzionale Lagrangiano  $\mathcal{L} : \mathcal{V} \times \mathcal{U} \times \mathcal{V} \rightarrow \mathbb{R}$  definito da

$$\mathcal{L}(y, u; p) = J(y, u) + (f, p)_{\mathcal{H}} + b(u, p) - a(y, p),$$

si può mostrare che il sistema (2.7) è dato da:

$$\begin{cases} \mathcal{L}_y = 0 \Rightarrow \text{trovare } p \in \mathcal{V} : a(\phi, p) = (Cy - z_d, C\phi)_{\mathcal{H}}, & \forall \phi \in \mathcal{V}, \\ \mathcal{L}_p = 0 \Rightarrow \text{trovare } y \in \mathcal{V} : a(y, \varphi) = b(u, \varphi) + (f, \varphi)_{\mathcal{H}}, & \forall \varphi \in \mathcal{V}, \\ \mathcal{L}_u = 0 \Rightarrow \text{trovare } u \in \mathcal{U}_{ad} : n(u, \psi) + b(\psi, p) = 0, & \forall \psi \in \mathcal{U}_{ad}. \end{cases}$$

Nella Sezione 2.4 verrà mostrato che anche per un problema di ottimizzazione di forma è possibile esprimere le condizioni di ottimalità sfruttando l'approccio basato sul problema aggiunto o sulla tecnica dei moltiplicatori di Lagrange. Prima di fare ciò occorre fornire una formulazione astratta di un problema di ottimizzazione di forma, analizzare l'esistenza di (almeno) una forma ottima (Sezione 2.2) e descrivere una possibile strategia per ottenere un insieme di forme ammissibili (Sezione 2.3).

## 2.2 Problemi di ottimizzazione di forma

Analogamente a quanto visto per i problemi di controllo ottimo, un problema di ottimizzazione di forma può essere scritto come la minimizzazione di un funzionale costo su un insieme di forme ammissibili  $\mathcal{O}_{ad}$ : determinare, se esiste, un dominio  $\widehat{\Omega} \in \mathcal{O}_{ad}$  tale che

$$\widehat{\Omega} = \arg \min_{\Omega \in \mathcal{O}_{ad}} J(\Omega).$$

In particolare:

1. il **dominio**  $\Omega \subset \mathbb{R}^n$  è la variabile di controllo, e  $\mathcal{O}_{ad} \subset \mathcal{O}$  è l'insieme delle forme ammissibili. Come discusso in seguito, l'insieme  $\mathcal{O}$  è costituito da una famiglia di forme sufficientemente regolari, in modo da garantire l'esistenza di (almeno) una forma ottima; eventuali vincoli di natura fisica (ad esempio, vincoli di volume) definiscono il suo sottoinsieme  $\mathcal{O}_{ad}$ ;
2. la **variabile di stato**  $u = u(\Omega)$  appartiene ad uno spazio di Hilbert  $\mathcal{V}(\Omega)$ , e soddisfa il seguente **problema di stato stazionario**:

$$\text{trovare } u(\Omega) \in \mathcal{V}(\Omega) \text{ tale che } A(\Omega)u(\Omega) = F(\Omega), \quad (2.8)$$

per un opportuno operatore  $A(\Omega) \in \mathcal{L}(\mathcal{V}(\Omega), \mathcal{V}(\Omega)')$  ed un funzionale  $F(\Omega) \in \mathcal{V}'(\Omega)$ , oppure il seguente **problema di stato non stazionario**:

$$\begin{aligned} &\text{per ogni } t \in (0, T), \text{ trovare } u(\Omega, t) \in \mathcal{V}(\Omega) \text{ tale che} \\ &\begin{cases} \frac{\partial u(\Omega, t)}{\partial t} + A(\Omega, t)u(\Omega, t) = F(\Omega, t), & t \in (0, T), \\ u(\Omega, 0) = u_0(\Omega); \end{cases} \end{aligned}$$

3. il **funzionale costo**, che rappresenta l'obiettivo da ottimizzare, è dato da

$$J(\Omega) = J(\Omega, u(\Omega)), \quad \forall \Omega \in \mathcal{O}_{ad};$$

nel caso più generale, si ha (per un problema di stato stazionario)

$$J(\Omega) = \int_{\Omega} j_{\Omega}(\mathbf{x}, u(\Omega), \nabla u(\Omega)) d\mathbf{x} + \int_{\partial\Omega} j_{\partial\Omega}(s, u(\Omega), \nabla u(\Omega)) ds,$$

con  $j_{\Omega} : \Omega \times \mathbb{R} \times \mathbb{R}^n \rightarrow \mathbb{R}$ ,  $j_{\partial\Omega} : \partial\Omega \times \mathbb{R} \times \mathbb{R}^n \rightarrow \mathbb{R}$ , oppure (per un problema di stato non stazionario)

$$\begin{aligned} J(\Omega) = &\int_0^T \int_{\Omega} j_{\Omega}(\mathbf{x}, t, u(\Omega, t), \nabla u(\Omega, t)) d\mathbf{x} dt + \int_0^T \int_{\partial\Omega} j_{\partial\Omega}(s, t, u(\Omega, t), \nabla u(\Omega, t)) ds dt + \\ &\int_{\Omega} j_{\Omega}^T(\mathbf{x}, T, u(\Omega, T), \nabla u(\Omega, T)) d\mathbf{x} + \int_{\partial\Omega} j_{\partial\Omega}^T(s, T, u(\Omega, T), \nabla u(\Omega, T)) ds, \end{aligned}$$

con  $j_{\Omega}, j_{\Omega}^T : \Omega \times \mathbb{R}^+ \times \mathbb{R} \times \mathbb{R}^n \rightarrow \mathbb{R}$ ,  $j_{\partial\Omega}, j_{\partial\Omega}^T : \partial\Omega \times \mathbb{R}^+ \times \mathbb{R} \times \mathbb{R}^n \rightarrow \mathbb{R}$ .

Rispetto ai problemi presentati nella sezione precedente, in cui la variabile di controllo è un coefficiente o una forzante dell'equazione di stato, la situazione che si presenta nei problemi di ottimizzazione di forma è più complicata, in quanto è il dominio stesso in cui viene definita l'equazione di stato ad essere soggetto ad ottimizzazione.

Il seguente Teorema garantisce l'esistenza di almeno una forma ottima, ed è il risultato più importante tra quelli contenuti in questo capitolo. Seppure enunciato per un generico problema di *shape optimization*, nel seguito se ne mostrerà in particolare l'applicazione ad un problema di ottimizzazione di forma con equazione di stato data da:

- (i) un problema di Poisson (Sezioni 2.2.4 e 2.3.1, per due diversi insiemi di domini ammissibili);
- (ii) un problema di Stokes stazionario (Sezione 3.2.2);
- (iii) un problema di Stokes non stazionario (Sezione 3.3.2).

**Teorema 2.4 (Teorema di esistenza [HP05, HM03]).** *Siano  $\{y_n\}_{n \geq 1} \subset \mathcal{V}(\Omega_n)$ ,  $\Omega_n \in \mathcal{O}$ , e  $y \in \mathcal{V}(\Omega)$ ,  $\Omega \in \mathcal{O}$ ; sia  $\tau$  una topologia di  $\mathcal{O}$  e si indichi con  $y_n \rightsquigarrow y$  una opportuna nozione di convergenza di  $y_n$  a  $y$ .*

*Si assuma inoltre che il problema di stato (2.8) sia ben posto, in modo che la sua soluzione al variare del dominio  $\Omega \in \mathcal{O}$  definisca una mappa  $u(\cdot)$  che associa a ciascun  $\Omega \in \mathcal{O}$  un elemento  $u(\Omega) \in \mathcal{V}(\Omega)$ :*

$$u : \Omega \mapsto u(\Omega) \in \mathcal{V}(\Omega), \quad \forall \Omega \in \mathcal{O}. \quad (P(\Omega))$$

*Sia  $\mathcal{G}$  il grafico della mappa  $u(\cdot)$  ristretta ad  $\mathcal{O}_{ad}$ , cioè  $\mathcal{G} = \{(\Omega, u(\Omega)), \forall \Omega \in \mathcal{O}_{ad}\}$ .*

*Si assuma infine la validità delle seguenti ipotesi:*

- (1) **compattezza di  $\mathcal{G}$ :** *per ogni successione  $\{(\Omega_n, u(\Omega_n))\}_{n \geq 1} \subset \mathcal{G}$ , esiste una sottosuccessione  $\{(\Omega_{n_k}, u(\Omega_{n_k}))\}_{n \geq 1}$  ed un elemento  $(\Omega, u(\Omega)) \in \mathcal{G}$  tale che*

$$\Omega_{n_k} \xrightarrow{\tau} \Omega, \quad u(\Omega_{n_k}) \rightsquigarrow u(\Omega), \quad \text{per } k \rightarrow \infty;$$

- (2) **semicontinuità inferiore di  $J$ :** *per ogni successione di domini  $\{\Omega_n\}_{n \geq 1} \subset \mathcal{O}$  e di funzioni  $\{y_n\}_{n \geq 1} \in \mathcal{V}(\Omega_n)$ , per ogni  $\Omega \in \mathcal{O}$  e  $y \in \mathcal{V}(\Omega)$ , tali che*

$$\Omega_n \xrightarrow{\tau} \Omega, \quad y_n \rightsquigarrow y \implies J(\Omega) \leq \liminf_{n \rightarrow \infty} J(\Omega_n).$$

*Allora, se  $J$  è limitato dal basso, esiste almeno un  $\hat{\Omega} \in \mathcal{O}_{ad}$  tale che*

$$\hat{\Omega} = \arg \min_{\Omega \in \mathcal{O}_{ad}} J(\Omega).$$

Il modo consueto di verificare la compattezza di  $\mathcal{G}$  si basa sui due seguenti passi:

- (1a) **compattezza di  $\mathcal{O}_{ad}$  in  $\mathcal{O}$ :** si dimostra che, per ogni successione  $\{\Omega_n\}_{n \geq 1} \subset \mathcal{O}_{ad}$ , esiste una sottosuccessione  $\{\Omega_{n_k}\}_{k \geq 1} \subset \mathcal{O}_{ad}$  ed un elemento  $\Omega \in \mathcal{O}_{ad}$  tale che  $\Omega_{n_k} \xrightarrow{\tau} \Omega$ , se  $k \rightarrow \infty$ ;
- (1b) **continuità di  $u(\Omega)$  rispetto al dominio:** si dimostra che la soluzione  $u(\Omega)$  di  $P(\Omega)$  dipende con continuità da variazioni di  $\Omega \in \mathcal{O}_{ad}$ , cioè dati  $\{\Omega_n\}_{n \geq 1} \subset \mathcal{O}_{ad}$ ,  $\Omega \in \mathcal{O}_{ad}$ , vale la seguente implicazione

$$\Omega_n \xrightarrow{\tau} \Omega \implies u(\Omega_n) \rightsquigarrow u(\Omega).$$

Gli strumenti necessari per soddisfare le ipotesi del Teorema 2.4 di esistenza verranno introdotti nelle prossime sezioni; in particolare:

- o nella Sezione 2.2.1 viene descritto come introdurre una nozione di convergenza sull'insieme di domini  $\mathcal{O}$  costituito dagli aperti di  $\mathbb{R}^n$  inclusi in un compatto fissato  $B$ , e ne viene enunciata una prima proprietà di compattezza;
- o nella Sezione 2.2.2 l'insieme delle forme ammissibili viene ristretto in modo che sia possibile dimostrare la continuità della soluzione del problema di stato rispetto al dominio;
- o nella Sezione 2.2.3 vengono mostrati due esempi di funzionali semicontinui inferiormente, di interesse applicativo.



### 2.2.1 Topologie ed una proprietà di compattezza

In questa sezione vengono presentate due nozioni di convergenza, nel senso delle funzioni caratteristiche e nel senso di Hausdorff, mostrando che per la seconda vale la proprietà di compattezza necessaria all'applicazione del Teorema 2.4 di esistenza.

Dato che, a differenza dei problemi di controllo ottimo, l'insieme  $\mathcal{O}$  delle "variabili di controllo" non è naturalmente dotato di una nozione di convergenza, occorre introdurre una topologia: una prima possibilità è rappresentata dalla convergenza nel senso delle funzioni caratteristiche, che può essere definita sotto l'ipotesi di misurabilità della successione di domini e del suo limite:

**Definizione 2.5 ([HP05]).** *Si consideri la successione  $\{E_k\}_{k \geq 1}$  e l'insieme  $E$ , misurabili in  $\mathbb{R}^n$ : si dice che  $E_k$  converge nel senso delle funzioni caratteristiche ad  $E$  ( $E_k \xrightarrow{\chi} E$ ) se, indicando con  $\chi_E$  la funzione caratteristica dell'insieme  $E$ , vale*

$$\chi_{E_k} \xrightarrow[k \rightarrow \infty]{} \chi_E \text{ in } L^p_{loc}(\mathbb{R}^n), \quad \forall p \in [1, \infty).$$

Restringendo la classe degli insiemi in esame ai soli compatti, una seconda possibilità è basata sul seguente concetto di distanza tra forme, nel senso di Hausdorff:

**Definizione 2.6 ([HP05]).** *Si consideri un compatto  $B \subset \mathbb{R}^n$ , fissato. Dati due compatti  $K_1, K_2 \subset \mathbb{R}^n$  tali che  $K_i \subset B$ , si pone*

$$\rho(K_1, K_2) = \sup_{x \in K_1} \inf_{y \in K_2} \|x - y\|,$$

e si indica con  $d^H(K_1, K_2)$  la distanza di Hausdorff tra  $K_1$  e  $K_2$ , la cui espressione è data da

$$d^H(K_1, K_2) = \max(\rho(K_1, K_2), \rho(K_2, K_1)).$$

Da essa discende una convergenza, detta di Hausdorff, anche per gli aperti inclusi nel compatto  $B$ , in base alla seguente:

**Definizione 2.7 ([HP05]).** *Si consideri il seguente insieme  $\mathcal{O}$ :*

$$\mathcal{O} = \{\Omega \text{ aperto} : \Omega \subset B\},$$

e sia  $\{\Omega_n\}_{n \geq 1} \subset \mathcal{O}$ ,  $\Omega \in \mathcal{O}$ : si dice che la successione  $\{\Omega_n\}_{n \geq 1}$  converge nel senso di Hausdorff a  $\Omega$  ( $\Omega_n \xrightarrow{H} \Omega$ ) se

$$d_H(\Omega_n, \Omega) = d^H(B \setminus \Omega_n, B \setminus \Omega) \xrightarrow[n \rightarrow \infty]{} 0.$$

La convergenza di Hausdorff è più conveniente rispetto a quella delle funzioni caratteristiche per l'applicazione del Teorema 2.4 di esistenza, in quanto si può dimostrare la seguente proprietà di compattezza:

**Proposizione 2.8 ([HP05]).** *Sia  $\{\Omega_n\}_{n \geq 1} \subset \mathcal{O}$ : allora esiste una sottosuccessione  $\{\Omega_{n_k}\}_{k \geq 1}$  ed un dominio  $\Omega \in \mathcal{O}$  tale che  $\Omega_{n_k} \xrightarrow{H} \Omega$ .*

Tuttavia questo risultato, da solo, non è sufficientemente forte da garantire l'applicabilità del Teorema 2.4 di esistenza, in quanto la sola convergenza nel senso di Hausdorff non implica necessariamente la continuità della soluzione del problema di stato rispetto al dominio: un possibile rimedio è basato sull'aggiunta di ulteriori ipotesi di regolarità sull'insieme delle forme ammissibili, come ad esempio quella di uniforme regolarità presentata nella prossima sezione.

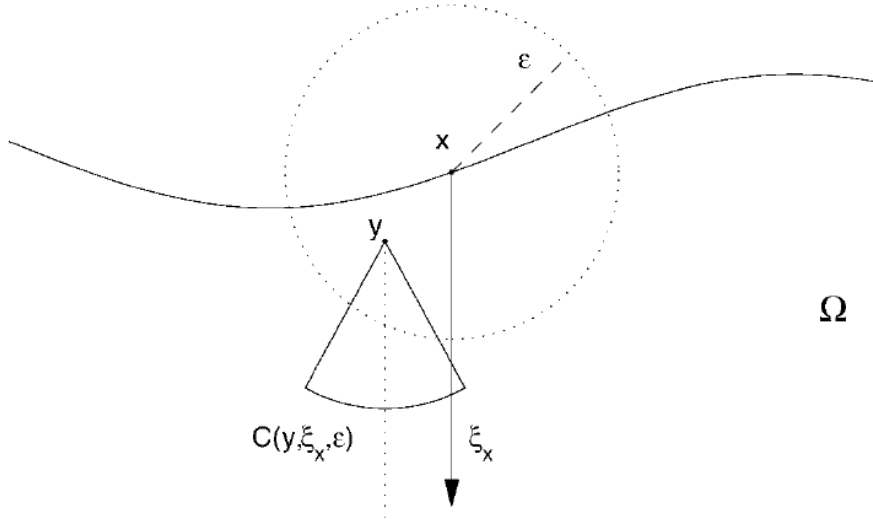


Figura 2.1: Proprietà dell' $\varepsilon$ -cono. Tratto da [HP05].

### 2.2.2 Domini uniformemente regolari

Una classe di forme che soddisfa un vincolo di ulteriore regolarità è quello dei domini che godono della proprietà dell' $\varepsilon$ -cono (o aperti uniformemente regolari), come indicato nella seguente Definizione (si veda anche la Figura 2.1):

**Definizione 2.9 ([HP05]).** *Dati  $y \in \mathbb{R}^n$ ,  $\xi$  versore ed  $\varepsilon > 0$ , si dice cono di vertice  $y$ , direzione  $\xi$  e dimensione  $\varepsilon$ , l'insieme  $C(y, \xi, \varepsilon)$  definito da*

$$C(y, \xi, \varepsilon) = \{z \in \mathbb{R}^n : (z - y) \cdot \xi \geq \cos(\varepsilon) |z - y|, 0 < |z - y| < \varepsilon\}.$$

Si dice che un aperto  $\Omega$  soddisfa la proprietà dell' $\varepsilon$ -cono se

$$\forall x \in \partial\Omega, \exists \xi = \xi_x \text{ versore} : C(y, \xi_x, \varepsilon) \subset \Omega \quad \forall y \in \bar{\Omega} \cap B_{r=\varepsilon}(x).$$

Si può mostrare che un aperto  $\Omega$  con frontiera limitata gode della proprietà dell' $\varepsilon$ -cono se e solo se è a bordo lipschitziano, cioè se giace interamente da un lato della sua frontiera, la quale è localmente il grafico di una funzione lipschitziana.

La proprietà di compattezza enunciata nella Proposizione 2.8 può essere ora rafforzata restringendo la classe dei domini ammissibili: imponendo che essi godano della proprietà dell' $\varepsilon$ -cono si riesce ad ottenere, oltre alla convergenza nel senso di Hausdorff, anche quella nel senso delle funzioni caratteristiche che, come verrà mostrato nella Sezione 2.2.4, si rivelerà fondamentale per la dimostrazione della *shape continuity* (in particolare, nella dimostrazione della validità delle condizioni di Dirichlet). Vale infatti il seguente risultato:

**Teorema 2.10 ([HP05]).** *Sia*

$$\mathcal{O}_\varepsilon = \{\Omega \text{ aperto} : \Omega \subset B, \Omega \text{ gode della proprietà dell}'\varepsilon\text{-cono}\}. \quad (2.9)$$

*Sia  $\{\Omega_n\}_{n \geq 1} \subset \mathcal{O}_\varepsilon$ : allora esiste una sottosuccessione  $\{\Omega_{n_k}\}_{k \geq 1}$  ed un dominio  $\Omega \in \mathcal{O}_\varepsilon$  tale che  $\Omega_n \xrightarrow{H} \Omega$  e  $\Omega_n \xrightarrow{\chi} \Omega$ . Valgono inoltre  $\bar{\Omega}_n \xrightarrow{H} \bar{\Omega}$  e  $\partial\Omega_n \xrightarrow{H} \partial\Omega$ .*

Infine, come anticipato nella Sezione 2.2, si può anche considerare un sottoinsieme di forme ammissibili  $\mathcal{O}_{ad}$ , determinato dall'introduzione di vincoli natura fisica; per garantire che, a valle dell'applicazione del Teorema 2.10, anche il dominio limite rispetti i vincoli bisogna richiedere che:

$$\mathcal{O}_{ad} \subset \mathcal{O}_\varepsilon \text{ è chiuso rispetto alla convergenza nel senso di Hausdorff.} \quad (2.10)$$

L'ipotesi (2.10) è rispettata per alcuni vincoli particolarmente comuni, quali

$$\begin{aligned} \mathcal{O}_{ad} &= \{\Omega \in \mathcal{O}_\varepsilon : |\Omega| = \bar{V}\}, \text{ oppure} \\ \mathcal{O}_{ad} &= \{\Omega \in \mathcal{O}_\varepsilon : |\Omega| \leq \bar{V}\} \text{ oppure} \\ \mathcal{O}_{ad} &= \{\Omega \in \mathcal{O}_\varepsilon : \Omega \supset K, \text{ dove } K \text{ è un compatto assegnato}\}. \end{aligned}$$

Infatti, nel primo e nel secondo caso, data una successione  $\{\Omega_n\} \subset \mathcal{O}_{ad}$ , il vincolo di volume è verificato anche al limite grazie alla convergenza nel senso delle funzioni caratteristiche, come enunciato nel Teorema 2.10. Il terzo caso è diretta conseguenza della prima parte della seguente Proposizione (che vale senza ipotesi di regolarità sul dominio):

**Proposizione 2.11 ([HP05]).** *Sia  $\{\Omega_n\}_{n \geq 1} \subset \mathcal{O}$ ,  $\Omega \in \mathcal{O}$  tali che  $\Omega_n \xrightarrow{H} \Omega$ .*

- (i) *Se  $K$  è un compatto incluso in  $\Omega$ , allora  $K$  è incluso in  $\Omega_n$  per  $n$  sufficientemente grande.*
- (ii) *In particolare, se  $\phi$  è una funzione a supporto compatto in  $\Omega$ , allora  $\phi$  è a supporto compatto anche in  $\Omega_n$ , per  $n$  sufficientemente grande.*

La seconda parte della Proposizione 2.11 sarà utilizzata, inoltre, in tutte le dimostrazioni della *shape continuity* per passare al limite nella formulazione debole del problema di stato.

### 2.2.3 Semicontinuità inferiore del funzionale costo

In questa sezione verrà discussa la semicontinuità inferiore di due classi di funzionali costo.

Si consideri una successione di domini  $\{\Omega_n\}_{n \geq 1} \subset \mathcal{O}$  e di funzioni  $\{y_n\}_{n \geq 1} \in \mathcal{V}(\Omega_n)$ , ed  $\Omega \in \mathcal{O}$  e  $y \in \mathcal{V}(\Omega)$ , tali che

$$\Omega_n \xrightarrow{H} \Omega, \quad y_n \rightsquigarrow y,$$

dove la convergenza  $\rightsquigarrow$  è definita per mezzo di opportune estensioni (indicate con  $\tilde{\cdot}$ ) di  $y_n$  ed  $y$  allo spazio  $\mathcal{V}(B)$ , essendo  $B$  il compatto che contiene tutti i domini ammissibili:

$$y_n \rightsquigarrow y \iff \tilde{y}_n \xrightarrow{\mathcal{V}(B)} \tilde{y}.$$

Nel caso generale in cui  $\mathcal{V}(\Omega) \subset H^m(\Omega)$ ,  $m \in \mathbb{N}$ , tali estensioni esistono grazie al Teorema di Calderón (si veda [Mar86, GK98]); per i problemi presentati in questa Tesi (in particolare nelle Sezioni 2.2.4, 2.3.1, 3.2.2 e 3.3.2) non sarà nemmeno necessario invocare tale risultato perché, scegliendo opportunamente il compatto  $B$  e sfruttando le condizioni di Dirichlet, l'estensione a zero in  $B \setminus \Omega$  sarà sufficiente.

Sotto queste ipotesi, per  $m \geq 1$  si può verificare che i seguenti funzionali costo sono sequenzialmente inferiormente semicontinui, rispetto alla convergenza  $\rightsquigarrow$ :

- Sia  $j_1 = j_1(\mathbf{x}, y, g) : B \times \mathbb{R} \times \mathbb{R}^n \rightarrow \mathbb{R}$  una funzione misurabile, continua rispetto a  $(y, g)$  per quasi ogni  $\mathbf{x}$ , e tale che esista una costante  $C$  per cui

$$|j_1(\mathbf{x}, y, g)| \leq C(1 + y^2 + |g|^2).$$

Particolarmente comune è la scelta  $j_1(\mathbf{x}, y, g) = (y - \bar{y})^2$ , essendo  $\bar{y} \in L^2(B)$  assegnato, oppure  $j_1(\mathbf{x}, y, g) = |g|^2$ . Il funzionale costo

$$J_1(\Omega, y) = \int_{\Omega} j_1(\mathbf{x}, y(\mathbf{x}), \nabla y(\mathbf{x})) d\mathbf{x}. \quad (2.11)$$

è ben definito in quanto le funzioni sotto il segno di integrale sono misurabili ed integrabili, e vale

$$|J_1(\Omega, y)| \leq C \int_{\Omega} 1 + y(\mathbf{x})^2 + |\nabla y(\mathbf{x})|^2 = C(|\Omega| + \|y\|_{H^1(\Omega)}) < +\infty. \quad (2.12)$$

Infine, il funzionale  $J$  è sequenzialmente inferiormente semicontinuo: per ulteriori dettagli si veda [HP05].

- Sia  $j_2 = j_2(g)$  continua, non negativa e convessa. Allora

$$J_2(\Omega, y) = \int_{\Omega} j_2(\nabla y(\mathbf{x})) d\mathbf{x} \quad (2.13)$$

è sequenzialmente inferiormente semicontinuo: per la dimostrazione si veda [Fuj88, GK98].

## 2.2.4 Un esempio: equazione di Poisson

In questa sezione viene mostrato come applicare i risultati precedenti all'analisi di un semplice problema di ottimizzazione di forma, in cui l'equazione di stato è data dal seguente problema di Poisson:

$$\begin{cases} -\Delta u(\Omega) = f, & \text{in } \Omega, \\ u(\Omega) = 0, & \text{su } \partial\Omega, \end{cases}$$

per  $f \in L^2(\mathbb{R}^n)$ , la cui formulazione debole è

$$\begin{aligned} &\text{trovare } u(\Omega) \in \mathcal{V}(\Omega) = H_0^1(\Omega) : \\ &a(u(\Omega), v) = F(v), \quad \forall v \in \mathcal{V}(\Omega), \end{aligned} \quad (2.14)$$

con  $a(u, v) = \int_{\Omega} \nabla u \cdot \nabla v$  e  $F(v) = \int_{\Omega} f v$ .

Anzitutto, grazie al Lemma di Lax-Milgram, si verifica che il problema di stato (2.14) gode delle proprietà di buona posizione richieste dal Teorema 2.4 di esistenza. In particolare, ponendo  $v = u(\Omega)$  nella forma debole, grazie alla disuguaglianza di Cauchy-Schwarz si ottiene

$$\int_{\Omega} |\nabla u(\Omega)|^2 = \int_{\Omega} f v \leq \|f\|_{L^2(\Omega)} \|u(\Omega)\|_{L^2(\Omega)}.$$

Applicando infine la disuguaglianza di Poincaré al secondo membro, ed indicando con  $\|\cdot\|_{\mathcal{V}(\Omega)}$  la norma equivalente  $\|v\|_{\mathcal{V}(\Omega)} = \|\nabla v\|_{L^2(\Omega)}$ , si ottiene la seguente stima di stabilità:

$$\|u(\Omega)\|_{\mathcal{V}(\Omega)} \leq C_u = C_P(\Omega) \|f\|_{L^2(\mathbb{R}^n)},$$

dove  $C_P(\Omega)$  è la costante di Poincaré in  $\mathcal{V}(\Omega)$ .

Nel seguente Teorema viene mostrato come applicare gli strumenti messi a disposizione dalle Sezioni da 2.2.1 a 2.2.3 ed il risultato astratto fornito dal Teorema 2.4 di esistenza ad un problema di ottimizzazione di forma con equazione di stato data da (2.14):

**Teorema 2.12.** *Sia  $\mathcal{O}_{ad}$  insieme di domini ammissibili, uniformemente regolari, che soddisfano la (2.10), e sia  $J(\Omega)$  dato da (2.11) oppure (2.13), in cui  $u(\Omega)$  è soluzione dell'equazione di Poisson (2.14). Allora esiste almeno un dominio  $\hat{\Omega} \in \mathcal{O}_{ad}$  che risolve il problema di ottimizzazione di forma*

$$\hat{\Omega} = \arg \min_{\Omega \in \mathcal{O}_{ad}} J(\Omega). \quad (2.15)$$

DIMOSTRAZIONE. Occorre verificare le ipotesi del Teorema 2.4 di esistenza, ed in particolare la *compattezza di  $\mathcal{G}$*  (che sarà conseguenza della *compattezza di  $\mathcal{O}_{ad}$  in  $\mathcal{O}_\varepsilon$*  e della *continuità di  $u(\Omega)$  rispetto al dominio*) e la *semicontinuità inferiore di  $J$* .

**Compattezza di  $\mathcal{O}_{ad}$  in  $\mathcal{O}_\varepsilon$ :** è conseguenza del Teorema 2.10, il quale, data una successione  $\{\Omega_n\}_{n \geq 1} \subset \mathcal{O}_{ad}$ , garantisce l'esistenza di una sottosuccessione (denominata sempre con indice  $n$ ) ed un dominio  $\Omega \in \mathcal{O}_\varepsilon$  tali che

$$\Omega_n \xrightarrow{H} \Omega.$$

Inoltre  $\Omega \in \mathcal{O}_{ad}$  grazie all'ipotesi (2.10) di chiusura.

**Continuità di  $u(\Omega)$  rispetto al dominio:** indicando con  $\{\Omega_n\}_{n \geq 1} \subset \mathcal{O}_{ad}$  la successione individuata in precedenza, e con  $\Omega \in \mathcal{O}_{ad}$  il suo limite, occorre dimostrare che

$$u(\Omega_n) \rightsquigarrow u(\Omega),$$

dove la convergenza  $\rightsquigarrow$  è definita da:

$$u(\Omega_n) \rightsquigarrow u(\Omega) \iff \tilde{u}(\Omega_n) \xrightarrow{\mathcal{V}(B)} \tilde{u}(\Omega),$$

avendo indicato con  $\tilde{u}(\Omega)$  l'estensione a zero in  $B \setminus \Omega$ . Si dimostra anzitutto che è possibile estrarre da  $\{\tilde{u}_n(\Omega)\}_{n \geq 1}$  una sottosuccessione convergente e che il limite di tale sottosuccessione è soluzione del problema di Poisson nel dominio  $\Omega$ :

1. *Estrazione di una sottosuccessione convergente:* sia  $u_n = u(\Omega_n)$ . Grazie alle condizioni di Dirichlet omogenee su  $\partial\Omega$  è possibile estendere a zero  $u_n$  in  $B \setminus \Omega_n$ ; tale estensione viene denominata  $\tilde{u}_n$ . Sostituendo gli integrali su  $\Omega_n$  con integrali su  $B$  nella dimostrazione della stima di stabilità si ottiene

$$\|\tilde{u}_n\|_{\mathcal{V}(B)} \leq C_u,$$

dove la costante  $C_u$  dipende da  $B$  tramite la costante di Poincaré  $C_P(B)$ , ma non da  $n$ . Di conseguenza è possibile estrarre una sottosuccessione, sempre denotata con  $\tilde{u}_n$ , tale che  $\tilde{u}_n \xrightarrow{\mathcal{V}(B)} \tilde{u}$  (Teorema di Banach-Alaoglu). Rimane da mostrare che  $u = \tilde{u}|_\Omega$  è soluzione del problema di Poisson in  $\Omega$ .

2.  *$u$  soddisfa le condizioni al contorno su  $\partial\Omega$ :* considerando l'identità

$$0 = \tilde{u}_n(\chi_B - \chi_{\Omega_n}), \tag{2.16}$$

per il Teorema di Rellich la convergenza debole  $\tilde{u}_n \xrightarrow{\mathcal{V}(B)} \tilde{u}$  in  $\mathcal{V}(B)$  implica la convergenza forte  $\tilde{u}_n \xrightarrow{L^2(B)} \tilde{u}$  in  $L^2(B)$ . Inoltre dal Teorema 2.10 discende che  $\Omega_n \xrightarrow{\chi} \Omega$ , e dalla Definizione 2.5 si ottiene che  $\chi_{\Omega_n} \xrightarrow{L^2(B)} \chi_\Omega$ . La (2.16) rimane quindi vera anche al limite in  $L^2(B)$ , da cui  $\tilde{u}|_{B \setminus \Omega} = 0$  quasi ovunque. Ciò è sufficiente per affermare che  $u = \tilde{u}|_\Omega$  appartiene a  $\mathcal{V}(\Omega)$  (si veda anche [HP05, Lemma 3.2.15]).

3.  *$u$  è soluzione dell'equazione di Poisson in  $\Omega$ :* è sufficiente passare al limite nella forma debole. Sia  $\phi \in \mathcal{D}(\Omega)$ : grazie alla Proposizione 2.11 essa è anche a supporto compatto in  $\Omega_n$  per  $n$  sufficientemente grande, e quindi  $\tilde{\phi}|_{\Omega_n}$  può essere usata come funzione test nella formulazione debole del problema di Poisson in  $\Omega_n$ . Infine, la convergenza debole  $\tilde{u}_n \xrightarrow{\mathcal{V}(B)} \tilde{u}$  giustifica la validità del seguente passaggio al limite

$$\int_{\Omega_n} \nabla u_n \cdot \nabla \tilde{\phi}|_{\Omega_n} = \int_B \nabla \tilde{u}_n \cdot \nabla \tilde{\phi} \xrightarrow{n \rightarrow \infty} \int_B \nabla \tilde{u} \cdot \nabla \tilde{\phi} = \int_\Omega \nabla u \cdot \nabla \phi,$$

da cui, per densità di  $\mathcal{D}(\Omega)$  in  $\mathcal{V}(\Omega)$ ,

$$\int_{\Omega} \nabla u \cdot \nabla v = \int_{\Omega} f v, \quad \forall v \in \mathcal{V}(\Omega).$$

4.  $u(\Omega_n) \rightsquigarrow u(\Omega)$ : infatti, ponendo  $v = u_n$  e passando al limite in

$$\int_B |\nabla \tilde{u}_n|^2 = \int_B f \tilde{u}_n,$$

si ottiene, per la convergenza debole in  $\mathcal{V}(B)$ ,

$$\int_B |\nabla \tilde{u}_n|^2 = \int_B f \tilde{u}_n \xrightarrow{n \rightarrow \infty} \int_B f \tilde{u} = \int_B |\nabla \tilde{u}|^2,$$

da cui

$$\|\tilde{u}_n\|_{\mathcal{V}(B)} \xrightarrow{n \rightarrow \infty} \|\tilde{u}\|_{\mathcal{V}(B)};$$

insieme alla convergenza debole nello stesso spazio, quest'ultima relazione implica la convergenza<sup>1</sup>  $u(\Omega_n) \rightsquigarrow u(\Omega)$ .

**Semicontinuità inferiore di  $J$ :** è conseguenza dei risultati riportati nella Sezione 2.2.3.

Le ipotesi del Teorema 2.4 di esistenza sono quindi tutte verificate (la limitatezza dal basso del funzionale (2.11) è conseguenza di (2.12), quella del funzionale (2.13) è conseguenza della non negatività della funzione integranda  $j_2(g)$ ), e dunque il problema di ottimizzazione di forma (2.15) ammette almeno una forma ottima.  $\square$

La famiglia  $\mathcal{O}_\varepsilon$  degli aperti uniformemente regolari non è l'unica che garantisce la validità di un risultato di esistenza simile a quello fornito dal Teorema 2.12; anzi, il vincolo di regolarità imposto dalla proprietà dell' $\varepsilon$ -cono è piuttosto stringente, e può essere rilassato introducendo la definizione di *capacità di un insieme*. Per ulteriori dettagli si veda [BB05]. Un caso particolare è dato dal risultato ottenuto da Sverák [Sve93], il quale garantisce la *shape continuity* in  $\mathbb{R}^2$  per l'insieme

$$\mathcal{O}_{cc}(M) = \{\Omega \subset B \subset \mathbb{R}^2 : \Omega \text{ aperto, } B \setminus \Omega \text{ ha al più } M \text{ componenti connesse.}\}.$$

La possibilità di imporre un diverso vincolo di regolarità verrà discussa nella Sezione 2.3.1 per mostrare l'esistenza di una forma ottima ottenuta mediante deformazione di un dominio di riferimento fissato.

I risultati ottenuti in questa sezione saranno generalizzati nella Sezione 3.2.2 quando il sistema fisico è governato da un problema di Stokes stazionario e nella Sezione 3.3.2 per un problema di Stokes non stazionario.

---

<sup>1</sup>Questo risultato permette di introdurre una nuova topologia sugli aperti di  $\mathbb{R}^n$ , che prende il nome di  $\gamma$ -convergenza, ed esprime la continuità rispetto al dominio della soluzione del problema di stato (2.14): con questa nozione, si può equivalentemente scrivere che  $\Omega_n \xrightarrow{H} \Omega$  implica  $\Omega_n \xrightarrow{\gamma} \Omega$ .

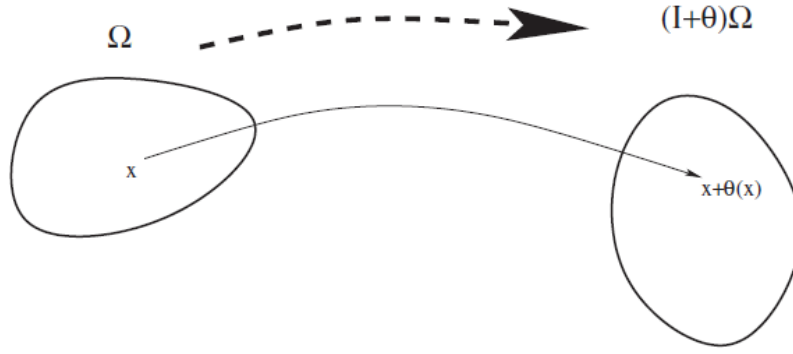


Figura 2.2: Mappa *perturbazione dell'identità*. Tratto da [All07].

## 2.3 Domini: descrizione e deformazione

Dopo aver mostrato, nelle sezioni precedenti, le ipotesi che garantiscono l'esistenza di una soluzione per un problema di ottimizzazione di forma, è opportuno sviluppare anche alcuni strumenti per la scrittura delle condizioni di ottimalità, che la caratterizzano. In particolare, con l'obiettivo di effettuare l'analisi di sensitività rispetto al dominio, è anzitutto necessario fornire il concetto di derivazione rispetto al dominio, che verrà introdotto nella Sezione 2.3.2. Esso è basato sulla generazione di domini tramite una famiglia di mappe dette *perturbazione dell'identità*: la *Free Form Deformation*, presentata nella Sezione 2.3.1, ne è un caso particolare.

Assegnato  $\Omega \subset \mathbb{R}^n$ , dominio di riferimento, si consideri il seguente spazio di diffeomorfismi in  $\mathbb{R}^n$ :

$$\mathcal{T} = \{\mathbf{T} : \mathbb{R}^n \rightarrow \mathbb{R}^n, (\mathbf{T} - \mathbf{I}) \in W^{1,\infty}(\mathbb{R}^n; \mathbb{R}^n), (\mathbf{T}^{-1} - \mathbf{I}) \in W^{1,\infty}(\mathbb{R}^n; \mathbb{R}^n)\},$$

e si introduca lo spazio  $\mathcal{O}_{\mathcal{T}}(\Omega)$  delle forme ottenute per deformazione di  $\Omega$ :

$$\mathcal{O}_{\mathcal{T}}(\Omega) = \{\Omega_o \text{ tali che } \Omega_o = \mathbf{T}(\Omega) \text{ per qualche } \mathbf{T} \in \mathcal{T}\}.$$

È possibile mostrare che la mappa  $\mathbf{T}$ , detta di *perturbazione dell'identità*, definita da

$$\mathbf{T} = \mathbf{I} + \boldsymbol{\theta} \text{ con } \boldsymbol{\theta} \in W^{1,\infty}(\mathbb{R}^n; \mathbb{R}^n)$$

appartiene, per il campo di spostamento  $\boldsymbol{\theta}$  scelto opportunamente, all'insieme  $\mathcal{T}$ . Vale infatti il seguente Lemma:

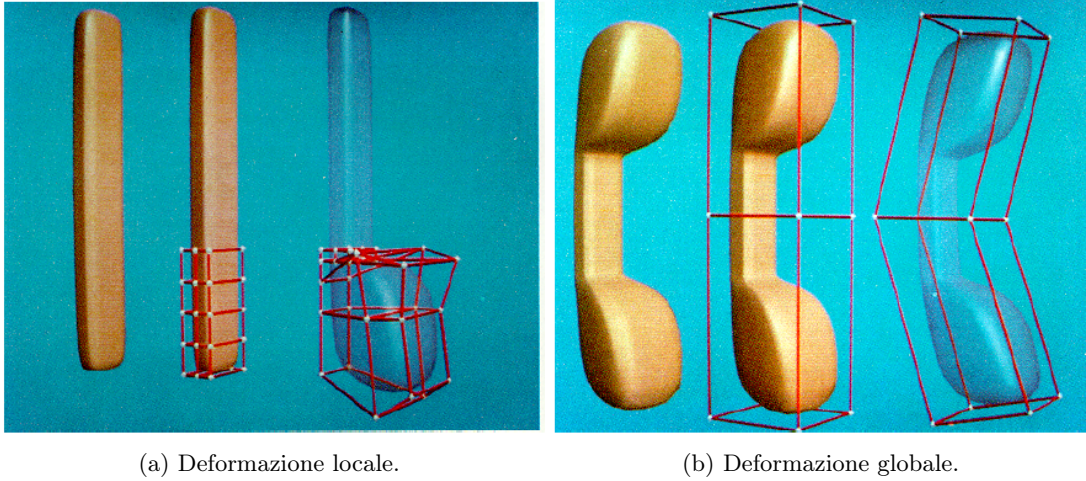
**Lemma 2.13 ([All07]).** *Sia  $\boldsymbol{\theta} \in W^{1,\infty}(\mathbb{R}^n; \mathbb{R}^n)$  tale che  $\|\boldsymbol{\theta}\|_{W^{1,\infty}(\mathbb{R}^n; \mathbb{R}^n)} < 1$ . Allora la mappa  $\mathbf{T} = \mathbf{I} + \boldsymbol{\theta} \in \mathcal{T}$ . In particolare, se  $\|\boldsymbol{\theta}\|_{W^{1,\infty}(\mathbb{R}^n; \mathbb{R}^n)} \leq 1 - \alpha$  per qualche  $\alpha > 0$ , allora*

$$\|\mathbf{T}^{-1} - \mathbf{I}\|_{W^{1,\infty}(\mathbb{R}^n; \mathbb{R}^n)} \leq \frac{1}{\alpha}.$$

In vista dell'applicazione al Teorema 2.4 di esistenza, è possibile introdurre su  $\mathcal{O}_{\mathcal{T}}(\Omega)$  la pseudo-distanza  $d_{\mathcal{T}}$  (essa verifica una versione debole della disuguaglianza triangolare), definita da

$$d_{\mathcal{T}}(\Omega, \Omega_o) = \inf_{\mathbf{T} \in \mathcal{T}: \mathbf{T}(\Omega) = \Omega_o} \{\|\mathbf{T} - \mathbf{I}\|_{W^{1,\infty}(\mathbb{R}^n; \mathbb{R}^n)} + \|\mathbf{T}^{-1} - \mathbf{I}\|_{W^{1,\infty}(\mathbb{R}^n; \mathbb{R}^n)}\},$$

e dimostrare il seguente risultato di compattezza, analogo al Teorema 2.10:



(a) Deformazione locale.

(b) Deformazione globale.

Figura 2.3: Esempi di deformazione secondo la mappa *Free Form Deformation*. Tratto da [SP86].

**Teorema 2.14** ([MS76, All07]). *Fissato  $r > 0$ , sia*

$$\mathcal{O}_{\mathcal{T}}(\Omega; r) = \{\Omega_o \in \mathcal{O}_{\mathcal{T}}(\Omega) : d_{\mathcal{T}}(\Omega, \Omega_o) \leq r\}.$$

*Sia  $\{\Omega_{o,n}\}_{n \geq 1} \subset \mathcal{O}_{\mathcal{T}}(\Omega; r)$ : allora esiste una sottosuccessione  $\{\Omega_{o,n_k}\}_{k \geq 1} \subset \mathcal{O}_{\mathcal{T}}(\Omega; r)$  ed un dominio  $\Omega_o \in \mathcal{O}_{\mathcal{T}}(\Omega; r)$  tale che  $\Omega_{o,n} \xrightarrow{H} \Omega_o$  e  $\Omega_{o,n} \xrightarrow{X} \Omega_o$ . Valgono inoltre  $\bar{\Omega}_{o,n} \xrightarrow{H} \bar{\Omega}_o$  e  $\partial\Omega_{o,n} \xrightarrow{H} \partial\Omega_o$ .*

### 2.3.1 La mappa *Free Form Deformation*

Da un punto di vista numerico, esistono numerose strategie per costruire una mappa di tipo *perturbazione dell'identità*. In generale, risulta conveniente che una mappa di questo tipo presenti le seguenti caratteristiche:

- regolarità;
- possibilità di descrivere un'ampia classe di domini, di forma anche complessa, con un numero ridotto di parametri di *design* (o gradi di libertà);
- possibilità di deformare i domini senza effettuare operazioni di *remeshing*.

Una possibile tecnica che soddisfa questi requisiti è quella basata su una parametrizzazione di tipo *Free Form Deformation*, originariamente introdotta in [SP86] e recentemente utilizzata nel contesto dei problemi di ottimizzazione di forma in [LR10, MQR11, AN08, LPQR11]. Questa tecnica consiste nel racchiudere la configurazione di riferimento (indeformata) in volume di controllo, sul quale è definita una griglia di punti di controllo (Figura 2.3(a)-(b), oggetto al centro), e permette di ottenere una configurazione deformata (Figura 2.3(a)-(b), oggetto a destra) tramite un'opportuna interpolazione degli spostamenti imposti sulla griglia di punti di controllo. I parametri che definiscono la mappa sono le deformazioni applicate ai punti di controllo (o ad un loro sottoinsieme, opportunamente scelto). Questa mappa è detta di tipo *free form* perché risulta indipendente dalla geometria del corpo in esame e può dunque essere applicata a forme arbitrariamente complesse mantenendo un numero ridotto di gradi di libertà, al contrario di altre parametrizzazioni più comuni (come, ad esempio, le superfici di Bézier) che, essendo costruite direttamente sulla frontiera del corpo da deformare, per descriverne accuratamente il comportamento necessitano di un numero crescente di gradi di libertà all'aumentare della complessità della forma. Inoltre, visto che si tratta di una trasformazione regolare, se gli spostamenti imposti sui



punti di controllo sono sufficientemente piccoli, la configurazione deformata gode delle stesse proprietà di regolarità del dominio di riferimento. Infine, dato che la trasformazione viene applicata a tutto il dominio in esame e non solo alla sua frontiera, è possibile applicare la deformazione anche alla mesh del dominio di riferimento, senza la necessità di effettuare costose operazioni di *remeshing*.

Tuttavia, essa gode anche di alcune limitazioni, particolarmente gravose soprattutto nel contesto della *computer graphics*, area in cui la *Free Form Deformation* è solitamente applicata. In particolare, nella sua versione originale, non è possibile cambiare la topologia della griglia dei punti di controllo, oppure addensare i punti di controllo solo in una certa regione; queste limitazioni sono state parzialmente superate da opportune estensioni (si parla a questo proposito di *Extended Free Form Deformation*; si veda [Coq90]). Per gli scopi di questa Tesi e dell'applicazione a problemi di ottimizzazione di forma queste limitazioni non sono troppo penalizzanti, dal momento che è possibile creare una griglia sufficientemente fitta ed abilitare solo un opportuno sottoinsieme di punti di controllo: la dimensionalità del problema di ottimizzazione corrisponderà dunque al numero dei soli punti di controllo abilitati. Una seconda limitazione è costituita dal fatto che non risulta immediato dedurre un legame semplice tra i valori di spostamento imposti ai punti di controllo e l'entità della deformazione a cui la forma è sottoposta: di nuovo, per l'applicazione a problemi di ottimizzazione di forma ciò non costituisce un limite perché, come si vedrà nella Sezione 4.4, l'aggiornamento dei valori di spostamento imposti è effettuato automaticamente dall'algoritmo di discesa. Infine, la scelta di quali punti di controllo siano da abilitare dipende essenzialmente dall'applicazione, come del resto accade anche con molte delle parametrizzazioni più usate.

Per introdurre la formulazione della mappa *Free Form Deformation* come perturbazione dell'identità, si consideri un dominio di riferimento  $\Omega \subset \mathbb{R}^3$ , fissato, e sia  $D = [x_1^{min}, x_1^{max}] \times [x_2^{min}, x_2^{max}] \times [x_3^{min}, x_3^{max}]$  un volume di controllo tale che  $\Omega \subset D$ . Allo scopo di definire la mappa rispetto alle coordinate  $[0, 1]^3$  ed utilizzare una base di *spline*, si consideri la mappa

$$\psi : D \rightarrow \hat{D} = [0, 1]^3, \quad \mathbf{x} \mapsto \hat{\mathbf{x}},$$

che trasforma il parallelepipedo di riferimento  $D$  nel cubo  $\hat{D}$ :

$$\psi(x_1, x_2, x_3) = \begin{bmatrix} \frac{x_1 - x_1^{min}}{x_1^{max} - x_1^{min}} \\ \frac{x_2 - x_2^{min}}{x_2^{max} - x_2^{min}} \\ \frac{x_3 - x_3^{min}}{x_3^{max} - x_3^{min}} \end{bmatrix}.$$

Considerando una griglia ordinata di  $(K + 1) \times (L + 1) \times (M + 1)$  punti di controllo su  $\hat{D}$

$$\hat{\mathbf{P}}_{k,l,m} = \begin{bmatrix} k/K \\ l/L \\ m/M \end{bmatrix}, \quad k = 0, \dots, K, l = 0, \dots, L, m = 0, \dots, M,$$

si indichi con  $\boldsymbol{\mu}_{k,l,m} \in \mathbb{R}^3$  il vettore spostamento a cui ciascun punto di controllo è sottoposto, e sia

$$\hat{\mathbf{P}}_{k,l,m} + \boldsymbol{\mu}_{k,l,m}$$

la posizione perturbata del punto di controllo  $(k, l, m)$ . È possibile che alcuni punti di controllo siano fissati o che ammettano spostamenti solo in alcune direzioni coordinate: per questo motivo il vettore  $\boldsymbol{\mu}$  indicherà, d'ora in poi, gli spostamenti scalari effettivamente abilitati: essi costituiscono i *gradi di libertà* della mappa *Free Form Deformation*.

Si consideri la seguente mappa  $\hat{\mathbf{T}}$  di deformazione del cubo  $\hat{D}$

$$\hat{\mathbf{T}} : \hat{D} \rightarrow \hat{D}_o(\boldsymbol{\mu}), \quad \hat{\mathbf{x}} \mapsto \hat{\mathbf{x}}_o(\boldsymbol{\mu}),$$

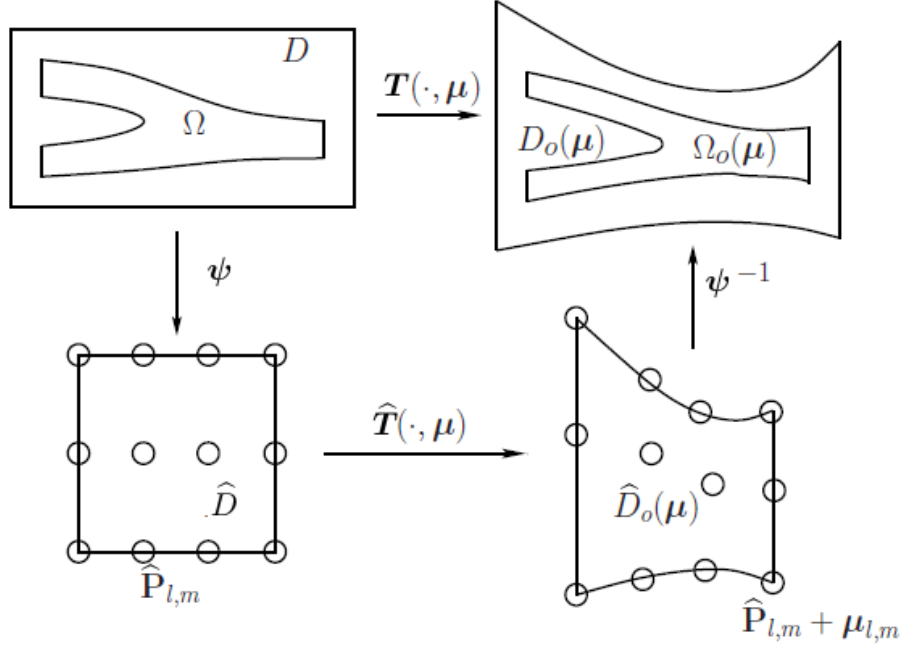


Figura 2.4: Diagramma della costruzione della mappa *Free Form Deformation*. Per semplificare la rappresentazione grafica si considera  $\Omega \subset \mathbb{R}^2$ . Tratto da [MQR11].

definita da

$$\hat{\mathbf{x}}_o(\boldsymbol{\mu}) = \hat{\mathbf{T}}(\hat{\mathbf{x}}; \boldsymbol{\mu}) = \sum_{k=0}^K \sum_{l=0}^L \sum_{m=0}^M b_{k,l,m}^{K,L,M}(\hat{\mathbf{x}}) [\hat{\mathbf{P}}_{k,l,m} + \boldsymbol{\mu}_{k,l,m}],$$

dove  $b_{k,l,m}^{K,L,M}(\hat{\mathbf{x}}) = b_{k,l,m}^{K,L,M}(\hat{x}_1, \hat{x}_2, \hat{x}_3)$  è il prodotto tensoriale dei polinomi di Bernstein monodimensionali

$$b_{k,l,m}^{K,L,M}(\hat{x}_1, \hat{x}_2, \hat{x}_3) = b_k^K(\hat{x}_1) b_l^L(\hat{x}_2) b_m^M(\hat{x}_3),$$

$$b_j^J(\hat{z}) = \binom{J}{j} (1 - \hat{z})^{J-j} \hat{z}^j.$$

Si definisce infine la mappa *Free Form Deformation*  $\mathbf{T} : D \rightarrow \mathbb{R}^3$ ,  $\mathbf{x} \mapsto \mathbf{x}_o(\boldsymbol{\mu}) = \mathbf{T}(\mathbf{x}; \boldsymbol{\mu})$ , come la composizione

$$\mathbf{T} = \boldsymbol{\psi}^{-1} \circ \hat{\mathbf{T}} \circ \boldsymbol{\psi},$$

e si pone

$$\Omega_o(\boldsymbol{\mu}) = \mathbf{T}(\Omega; \boldsymbol{\mu}),$$

indicando con  $\Omega_o(\boldsymbol{\mu})$  il dominio che si ottiene a partire dal dominio di riferimento  $\Omega$  tramite la mappa *Free Form Deformation* imponendo uno spostamento  $\boldsymbol{\mu}$  ai punti di controllo abilitati. La costruzione della mappa è schematizzata in Figura 2.4.

Sostituendo le espressioni ricavate in precedenza, sfruttando il fatto che i nodi di controllo sono posti in una griglia cartesiana equispaziata e ricordando che vale  $\sum_{j=0}^J b_j^J(\hat{z}) = 1$  (partizione dell'unità), è possibile esprimere  $\mathbf{T}$  come perturbazione dell'identità (almeno per  $\boldsymbol{\mu}$  sufficientemente piccolo, come verrà mostrato nel Teorema 2.15)

$$\mathbf{T}(\mathbf{x}; \boldsymbol{\mu}) = \mathbf{x} + \boldsymbol{\theta}(\mathbf{x}; \boldsymbol{\mu}), \quad (2.17)$$

avendo posto

$$\boldsymbol{\theta}(\mathbf{x}; \boldsymbol{\mu}) = \sum_{k=0}^K \sum_{l=0}^L \sum_{m=0}^M b_{k,l,m}^{K,L,M}(\boldsymbol{\psi}(\mathbf{x})) \mathfrak{D} \boldsymbol{\mu}_{k,l,m}, \quad (2.18)$$

ed essendo  $\mathfrak{D} = \mathfrak{D}(D)$  la matrice data da

$$\mathfrak{D} = \begin{bmatrix} D_1 & 0 & 0 \\ 0 & D_2 & 0 \\ 0 & 0 & D_3 \end{bmatrix} = \begin{bmatrix} x_1^{max} - x_1^{min} & 0 & 0 \\ 0 & x_2^{max} - x_2^{min} & 0 \\ 0 & 0 & x_3^{max} - x_3^{min} \end{bmatrix}.$$

In particolare si può esprimere il gradiente della mappa *Free Form Deformation* nel seguente modo:

$$\nabla \mathbf{T}(\mathbf{x}; \boldsymbol{\mu}) = I + \mathfrak{D} \left( \sum_{k,l,m} \boldsymbol{\mu}_{k,l,m} \otimes \nabla b_{k,l,m}^{K,L,M}(\boldsymbol{\psi}(\mathbf{x})) \right) \mathfrak{D}^{-1},$$

dove

$$\nabla b_{k,l,m}^{K,L,M}(\hat{x}_1, \hat{x}_2, \hat{x}_3) = \begin{bmatrix} K[b_{k-1}^{K-1}(\hat{x}_1) - b_k^{K-1}(\hat{x}_1)]b_l^L(\hat{x}_2)b_m^M(\hat{x}_3) \\ L[b_{l-1}^{L-1}(\hat{x}_2) - b_l^{L-1}(\hat{x}_2)]b_k^K(\hat{x}_1)b_m^M(\hat{x}_3) \\ M[b_{m-1}^{M-1}(\hat{x}_3) - b_m^{M-1}(\hat{x}_3)]b_k^K(\hat{x}_1)b_l^L(\hat{x}_2) \end{bmatrix}. \quad (2.19)$$

Questa espressione verrà utilizzata nella dimostrazione del prossimo Teorema 2.15 e nella Sezione 4.4, in cui si mostrerà che l'effettivo impiego numerico della mappa *Free Form Deformation*.

### Un risultato di esistenza per problemi di ottimizzazione di forma

Se il vettore dei gradi di libertà  $\boldsymbol{\mu}$  viene scelto opportunamente è possibile mostrare che la mappa *Free Form Deformation* risulta una perturbazione dell'identità, come dimostrato nel seguente Teorema 2.15; inoltre, è possibile applicare il risultato di compattezza fornito dal Teorema 2.14 e affermare dunque, grazie al Teorema 2.4 di esistenza, che deformando  $\Omega$  tramite la mappa *Free Form Deformation* esiste almeno una forma ottima per il problema di ottimizzazione di forma con equazione di stato data dal problema di Poisson (2.14).

**Teorema 2.15.** *Sia  $R > 0$  e si consideri la seguente famiglia di domini ottenuti per deformazione del dominio di riferimento  $\Omega$  tramite la mappa *Free Form Deformation* (2.17)<sup>2</sup>*

$$\mathcal{O}^{FFD}(\Omega, R) = \{\Omega_o : \Omega_o = \mathbf{T}(\Omega; \boldsymbol{\mu}) \text{ per qualche } \boldsymbol{\mu} \text{ tale che } \sum_{k,l,m} \|\boldsymbol{\mu}_{k,l,m}\|_2 \leq R\}.$$

Allora, esiste un  $\bar{R} > 0$  tale che tutti i domini appartenenti alla famiglia  $\mathcal{O}^{FFD}(\Omega, \bar{R})$  possono essere ottenuti per perturbazione dell'identità del dominio di riferimento  $\Omega$ , cioè appartengono all'insieme  $\mathcal{O}_{\mathcal{T}}(\Omega)$ ; in particolare, si può prendere

$$\bar{R} = \bar{R}(D, K, L, M; \alpha) = \frac{D_{min}}{D_{max}} \frac{1 - \alpha}{D_{min} + 2\sqrt{K^2 + L^2 + M^2}}, \quad (2.20)$$

dove

$$D_{min} = \min_{i=1,2,3} D_i, \quad D_{max} = \max_{i=1,2,3} D_i, \quad \alpha \in (0, 1) \text{ arbitrario.}$$

Se inoltre l'insieme dei domini ammissibili  $\mathcal{O}_{ad}^{FFD}(\Omega)$  è tale che

$$\mathcal{O}_{ad}^{FFD}(\Omega) \subset \mathcal{O}^{FFD}(\Omega, \bar{R}) \text{ è chiuso rispetto alla convergenza nel senso di Hausdorff,}$$

e  $J(\Omega_o)$  è dato da (2.11) o (2.13), in cui  $u = u(\Omega_o)$  è soluzione dell'equazione di Poisson (2.14), allora esiste almeno un dominio  $\hat{\Omega}$  che risolve il problema di ottimizzazione di forma

$$\hat{\Omega} = \arg \min_{\Omega_o \in \mathcal{O}_{ad}^{FFD}(\Omega)} J(\Omega_o). \quad (2.21)$$

<sup>2</sup>Nel seguito  $\|\cdot\|_2$  indicherà, a seconda del contesto, la norma euclidea di un vettore o la norma matriciale da essa indotta.

DIMOSTRAZIONE. Per mostrare che tutti i domini di  $\mathcal{O}^{FFD}(\Omega, \bar{R})$  sono perturbazioni dell'identità basta verificare che la scelta della costante  $\bar{R}$  in (2.20) assicura la validità dell'ipotesi

$$\|\boldsymbol{\theta}(\cdot; \boldsymbol{\mu})\|_{W^{1,\infty}(\mathbb{R}^n; \mathbb{R}^n)} = \sup_{\mathbf{x} \in \mathbb{R}^n} \|\boldsymbol{\theta}(\mathbf{x}; \boldsymbol{\mu})\|_2 + \sup_{\mathbf{x} \in \mathbb{R}^n} \|\nabla \boldsymbol{\theta}(\mathbf{x}; \boldsymbol{\mu})\|_2 < 1$$

presente nel Lemma 2.13. Infatti, si consideri il campo di trasporto  $\boldsymbol{\theta}(\mathbf{x}; \boldsymbol{\mu})$  associato alla mappa *Free Form Deformation*; valgono allora le seguenti maggiorazioni:

- sfruttando il fatto che le funzioni  $b_j^J(\hat{z})$ ,  $j = 0, \dots, J$  sono positive e formano una partizione dell'unità, si ha che

$$\begin{aligned} \|\boldsymbol{\theta}(\mathbf{x}; \boldsymbol{\mu})\|_2 &= \left\| \sum_{k,l,m} b_{k,l,m}^{K,L,M}(\boldsymbol{\psi}(\mathbf{x})) \mathfrak{D} \boldsymbol{\mu}_{k,l,m} \right\|_2 \leq \\ &\leq \sum_{k,l,m} \left| b_{k,l,m}^{K,L,M}(\boldsymbol{\psi}(\mathbf{x})) \right| \|\mathfrak{D} \boldsymbol{\mu}_{k,l,m}\|_2 \leq \sum_{k,l,m} \|\mathfrak{D} \boldsymbol{\mu}_{k,l,m}\|_2 \leq \\ &\leq \|\mathfrak{D}\|_2 \left\{ \sum_{k,l,m} \|\boldsymbol{\mu}_{k,l,m}\|_2 \right\} = D_{max} \sum_{k,l,m} \|\boldsymbol{\mu}_{k,l,m}\|_2; \end{aligned} \quad (2.22)$$

- dalla disuguaglianza di Cauchy-Schwarz si ottiene

$$\begin{aligned} \|\mathbf{a} \otimes \mathbf{b}\|_2 &= \sup_{\mathbf{c} \neq \mathbf{0}} \frac{\|(\mathbf{a} \otimes \mathbf{b})\mathbf{c}\|_2}{\|\mathbf{c}\|_2} = \sup_{\mathbf{c} \neq \mathbf{0}} \frac{\|\mathbf{b} \cdot \mathbf{c}\| \|\mathbf{a}\|_2}{\|\mathbf{c}\|_2} \leq \\ &\leq \sup_{\mathbf{c} \neq \mathbf{0}} \frac{\|\mathbf{a}\|_2 \|\mathbf{b}\|_2 \|\mathbf{c}\|_2}{\|\mathbf{c}\|_2} \leq \|\mathbf{a}\|_2 \|\mathbf{b}\|_2. \end{aligned}$$

Dalla submoltiplicatività della norma matriciale indotta e dalla precedente relazione, si ricava che

$$\begin{aligned} \|\nabla \boldsymbol{\theta}(\mathbf{x}; \boldsymbol{\mu})\|_2 &= \left\| \mathfrak{D} \left( \sum_{k,l,m} \boldsymbol{\mu}_{k,l,m} \otimes \nabla b_{k,l,m}^{K,L,M}(\boldsymbol{\psi}(\mathbf{x})) \right) \mathfrak{D}^{-1} \right\|_2 \leq \\ &\leq \|\mathfrak{D}\|_2 \left( \sum_{k,l,m} \left\| \boldsymbol{\mu}_{k,l,m} \otimes \nabla b_{k,l,m}^{K,L,M}(\boldsymbol{\psi}(\mathbf{x})) \right\|_2 \right) \|\mathfrak{D}^{-1}\|_2 \leq \\ &\leq \frac{D_{max}}{D_{min}} \sum_{k,l,m} \|\boldsymbol{\mu}_{k,l,m}\|_2 \left\| \nabla b_{k,l,m}^{K,L,M}(\boldsymbol{\psi}(\mathbf{x})) \right\|_2. \end{aligned}$$

Inoltre, l'espressione (2.19) del gradiente del prodotto tensoriale di polinomi di Bernstein permette di calcolare, sfruttando nuovamente positività e proprietà di partizione dell'unità, la seguente maggiorazione:

$$\left\| \nabla b_{k,l,m}^{K,L,M}(\boldsymbol{\psi}(\mathbf{x})) \right\|_2 \leq 2\sqrt{K^2 + L^2 + M^2}. \quad (2.23)$$

Sfruttando le maggiorazioni (2.22)-(2.23) si ottiene dunque

$$\|\boldsymbol{\theta}(\cdot; \boldsymbol{\mu})\|_{W^{1,\infty}(\mathbb{R}^n; \mathbb{R}^n)} \leq D_{max} \sum_{k,l,m} \|\boldsymbol{\mu}_{k,l,m}\|_2 \left( \frac{D_{min} + 2\sqrt{K^2 + L^2 + M^2}}{D_{min}} \right),$$

da cui, grazie alla (2.20), si ottiene

$$\|\boldsymbol{\theta}(\cdot; \boldsymbol{\mu})\|_{W^{1,\infty}(\mathbb{R}^n; \mathbb{R}^n)} \leq 1 - \alpha. \quad (2.24)$$

Il Lemma 2.13 permette infine di affermare che la trasformazione così ottenuta è una perturbazione dell'identità.

Per dimostrare l'esistenza di una soluzione per il problema di ottimizzazione di forma (2.21), occorre verificare le ipotesi del Teorema 2.4 di esistenza, ed in particolare la *compattezza di  $\mathcal{G}$*  (che risulterà a sua volta conseguenza della *compattezza di  $\mathcal{O}_{ad}^{FFD}(\Omega)$  in  $\mathcal{O}^{FFD}(\Omega, \bar{R})$*  e della *continuità di  $u(\Omega)$  rispetto al dominio*) e la *semicontinuità inferiore del funzionale costo  $J$* . Se esiste un  $r > 0$  tale che

$$d_{\mathcal{T}}(\Omega, \Omega_o) \leq r \quad \forall \Omega_o \in \mathcal{O}_{ad}^{FFD}(\Omega),$$

si può ripercorrere la dimostrazione del Teorema 2.12, a meno di sostituire il risultato di compattezza dovuto al Teorema 2.10 (valido per gli aperti uniformemente regolari) con quello del Teorema 2.14 (valido per le perturbazioni dell'identità, per un opportuno  $r > 0$ ). Infine, tale  $r$  esiste e risulta pari a  $r = 1 - \alpha + \frac{1}{\alpha}$ , come si ricava da (2.24), dalla seconda parte del Lemma 2.13 e dalla definizione della pseudo-distanza  $d_{\mathcal{T}}$ .  $\square$

È possibile applicare questo risultato anche a problemi di ottimizzazione di forma più generali, come quelli che verranno mostrati nelle Sezioni 3.2 e 3.3, modificando la dimostrazione della *continuità della soluzione del problema di stato rispetto al dominio* e della *semicontinuità inferiore del funzionale costo*, come indicato nelle Sezioni 3.2.2 e 3.3.2, rispettivamente.

### 2.3.2 Derivata di forma

La famiglia delle mappe di tipo *perturbazione dell'identità* permette di introdurre in modo semplice il seguente concetto di derivabilità rispetto alla forma, necessario per la scrittura delle condizioni di ottimalità secondo l'approccio presentato nella Sezione 2.4.1. In particolare, per quanto riguarda la mappa *Free Form Deformation*, risulterà inoltre possibile collegare la derivazione rispetto alla forma con il gradiente rispetto ai gradi di libertà  $\boldsymbol{\mu}$ , come verrà mostrato nella Sezione 4.4.1.

**Definizione 2.16 ([HP05]).** *Sia  $J : \mathcal{O}_{\mathcal{T}}(\Omega) \rightarrow \mathbb{R}$  e siano  $\mathbf{T}_t = \mathbf{I} + t\boldsymbol{\theta}$ ,  $\Omega_t = \mathbf{T}_t(\Omega)$ : si definisce derivata euleriana di  $J$ , in  $\Omega$  e in direzione  $\boldsymbol{\theta}$ , il seguente limite*

$$dJ(\Omega; \boldsymbol{\theta}) = \lim_{t \rightarrow 0} \frac{J(\Omega_t) - J(\Omega)}{t}.$$

*Si dice che  $J$  è derivabile rispetto al dominio in  $\Omega$  se esso ammette derivata euleriana in  $\Omega$  per ogni  $\boldsymbol{\theta} \in \mathcal{D}(\mathbb{R}^n, \mathbb{R}^n)$  e la mappa*

$$\boldsymbol{\theta} \mapsto dJ(\Omega; \boldsymbol{\theta})$$

*è lineare e continua. In tal caso si ha*

$$dJ(\Omega; \boldsymbol{\theta}) = \langle \nabla J(\Omega), \boldsymbol{\theta} \rangle_{\mathcal{D}'(\mathbb{R}^n, \mathbb{R}^n) \times \mathcal{D}(\mathbb{R}^n, \mathbb{R}^n)},$$

*e l'elemento  $\nabla J$  prende il nome di shape gradient.*

Grazie a questo concetto è possibile introdurre alcune regole di derivazione di integrali rispetto al dominio (o rispetto al bordo del dominio) su cui sono definiti. Vale in particolare il seguente risultato, in cui si suppone che le funzioni integrande siano definite su  $\mathbb{R}^n$  (cioè, non dipendano dal dominio di integrazione):

**Teorema 2.17 ([All07]).** *Sia  $\Omega$  un dominio regolare.*

1. *Data  $f \in W^{1,1}(\mathbb{R}^n)$ , sia  $J : \mathcal{O}_{\mathcal{T}}(\Omega) \rightarrow \mathbb{R}$  il funzionale definito da*

$$J(\Omega_o) = \int_{\Omega_o} f(\mathbf{x}) d\mathbf{x}.$$

Allora  $J(\Omega_o)$  è derivabile in  $\Omega$ , in direzione  $\boldsymbol{\theta}$ , e vale

$$dJ(\Omega; \boldsymbol{\theta}) = \int_{\partial\Omega} f \boldsymbol{\theta} \cdot \mathbf{n} \, dx, \quad \forall \boldsymbol{\theta} \in W^{1,\infty}(\mathbb{R}^n; \mathbb{R}^n),$$

dove  $\mathbf{n}$  è la normale esterna alla superficie  $\partial\Omega$ .

2. Data  $f \in W^{2,1}(\mathbb{R}^n)$ , sia  $J : \mathcal{O}_{\mathcal{T}}(\Omega) \rightarrow \mathbb{R}$  il funzionale definito da

$$J(\Omega_o) = \int_{\partial\Omega_o} f(s) \, ds.$$

Allora  $J(\Omega_o)$  è derivabile in  $\Omega$ , in direzione  $\boldsymbol{\theta}$ , e vale

$$dJ(\Omega; \boldsymbol{\theta}) = \int_{\partial\Omega} \left( \frac{\partial f}{\partial \mathbf{n}} + Hf \right) \boldsymbol{\theta} \cdot \mathbf{n} \, ds, \quad \forall \boldsymbol{\theta} \in C^1(\mathbb{R}^n; \mathbb{R}^n),$$

dove  $\mathbf{n}$  è la normale esterna alla superficie  $\partial\Omega$  e  $H = \operatorname{div} \mathbf{n}$  è la curvatura media della superficie  $\partial\Omega$ .

Tuttavia, questo risultato non è sufficientemente generale per essere applicato ad un problema di ottimizzazione di forma; infatti, come indicato nella Sezione 2.2, la funzione integranda che compare nel funzionale costo  $J(\Omega_o)$  di un problema di ottimizzazione di forma dipende necessariamente dal dominio  $\Omega_o$ , tramite la soluzione del problema di stato. È necessario dunque estendere il risultato del Teorema precedente al caso in cui le funzioni  $f$  dipendano dal dominio  $\Omega_o$ : ciò è possibile grazie all'introduzione del concetto di *shape derivative*.

**Definizione 2.18 ([HM03]).** Sia  $y = y(\Omega, \mathbf{x}) \in W(\Omega)$ , con  $W(\Omega)$  opportuno spazio di Sobolev, e siano  $\mathbf{T}_t = \mathbf{I} + t\boldsymbol{\theta}$ ,  $\Omega_t = \mathbf{T}_t(\Omega)$ .

Si definisce material derivative l'elemento  $\dot{y} = \dot{y}(\Omega) \in W(\Omega)$  tale che

$$\dot{y}(\Omega) = \lim_{t \rightarrow 0} \frac{y(\Omega_t) \circ \mathbf{T}_t(\Omega) - y(\Omega)}{t}.$$

Si definisce shape derivative l'elemento  $y' = y'(\Omega) \in W(\Omega)$  tale che

$$y'(\Omega) = \dot{y}(\Omega) - \boldsymbol{\theta} \cdot \nabla y(\Omega).$$

L'estensione del Teorema 2.17 è quindi fornita dal seguente risultato:

**Teorema 2.19 ([HP05]).** Sia  $\Omega$  un dominio regolare.

1. Dato  $y : \mathcal{O}_{\mathcal{T}}(\Omega) \rightarrow L^1(\mathbb{R}^n)$ , tale che  $y(\Omega) \in W^{1,1}(\mathbb{R}^n)$ , sia  $J : \mathcal{O}_{\mathcal{T}}(\Omega) \rightarrow \mathbb{R}$  il funzionale definito da

$$J(\Omega_o) = \int_{\Omega_o} y(\Omega_o, \mathbf{x}) \, d\mathbf{x}.$$

Allora  $J(\Omega_o)$  è derivabile in  $\Omega$ , in direzione  $\boldsymbol{\theta}$ , e vale

$$dJ(\Omega; \boldsymbol{\theta}) = \int_{\Omega} y'(\Omega) \, d\mathbf{x} + \int_{\partial\Omega} y(\Omega) \boldsymbol{\theta} \cdot \mathbf{n} \, dx, \quad \forall \boldsymbol{\theta} \in W^{1,\infty}(\mathbb{R}^n; \mathbb{R}^n),$$

dove  $\mathbf{n}$  è la normale esterna alla superficie  $\partial\Omega$ .

2. Dato  $y : \mathcal{O}_{\mathcal{T}}(\Omega) \rightarrow L^1(\mathbb{R}^n)$ , tale che  $y(\Omega) \in W^{2,1}(\mathbb{R}^n)$ , sia  $J : \mathcal{O}_{\mathcal{T}}(\Omega) \rightarrow \mathbb{R}$  il funzionale definito da

$$J(\Omega_o) = \int_{\partial\Omega_o} y(\Omega_o, s) \, ds.$$

Allora  $J(\Omega_o)$  è derivabile in  $\Omega$ , in direzione  $\boldsymbol{\theta}$ , e vale

$$dJ(\Omega; \boldsymbol{\theta}) = \int_{\partial\Omega} y'(\Omega) ds + \int_{\partial\Omega} \left( \frac{\partial y(\Omega)}{\partial \mathbf{n}} + Hy(\Omega) \right) \boldsymbol{\theta} \cdot \mathbf{n} ds, \quad \forall \boldsymbol{\theta} \in C^1(\mathbb{R}^n; \mathbb{R}^n),$$

dove  $\mathbf{n}$  è la normale esterna alla superficie  $\partial\Omega$  e  $H = \operatorname{div} \mathbf{n}$  è la curvatura media della superficie  $\partial\Omega$ .

Sotto opportune ipotesi (si veda ad esempio [DZ01]), si può inoltre dimostrare che, per una ampia classe di funzionali  $J$ , vale il seguente Teorema di struttura di Hadamard:

**Teorema 2.20 (Teorema di struttura [DZ01], versione semplificata).** *Siano  $\Omega$  un dominio e  $J : \mathcal{O}_{\mathcal{T}}(\Omega) \rightarrow \mathbb{R}$  un funzionale costo, sufficientemente regolari. Allora esiste una distribuzione scalare  $g(\Omega)$  con supporto in  $\partial\Omega$  tale che  $g(\Omega) \in \mathcal{D}'(\partial\Omega)$  e per ogni  $\boldsymbol{\theta} \in \mathcal{D}(\mathbb{R}^n, \mathbb{R}^n)$  vale*

$$dJ(\Omega; \boldsymbol{\theta}) = \langle g(\Omega), \boldsymbol{\theta} \cdot \mathbf{n} \rangle_{\mathcal{D}'(\partial\Omega) \times \mathcal{D}(\partial\Omega)}.$$

Inoltre, se  $g(\Omega) \in L^1(\partial\Omega)$ , vale la seguente formula di Hadamard:

$$dJ(\Omega; \boldsymbol{\theta}) = \int_{\partial\Omega} g(\Omega) \boldsymbol{\theta} \cdot \mathbf{n}. \quad (2.25)$$

Questi risultati saranno utilizzati in seguito per ricavare le condizioni di ottimalità di un problema di ottimizzazione di forma, che consentono di trovare l'espressione di  $g(\Omega)$  in funzione della soluzione del problema di stato e della soluzione di un opportuno *problema aggiunto*.

### 2.3.3 Calcolo della *shape derivative* per l'equazione di Poisson

In questa sezione viene mostrato come utilizzare gli strumenti di *shape analysis* introdotti nella Sezione 2.3.2 per ricavare la *shape derivative* della soluzione  $u(\Omega)$  di un'equazione di Poisson: è infatti necessario scrivere la *shape derivative* del problema di stato per formulare le condizioni di ottimalità secondo l'approccio presentato nella Sezione 2.4.1.

Si consideri il seguente problema di Poisson:

$$\begin{cases} -\Delta u(\Omega) = f, & \text{in } \Omega, \\ u(\Omega) = g, & \text{su } \partial\Omega, \end{cases} \quad (2.26)$$

per  $f \in L^2(\mathbb{R}^n)$ ,  $g \in H^2(\mathbb{R}^n)$ , la cui formulazione debole è

$$\begin{aligned} &\text{trovare } u(\Omega) - g \in \mathcal{V}(\Omega) = H_0^1(\Omega) : \\ &a(u(\Omega), v) = F(v), \quad \forall v \in \mathcal{V}(\Omega), \end{aligned}$$

con  $a(u, v) = \int_{\Omega} \nabla u \cdot \nabla v$  e  $F(v) = \int_{\Omega} f v$ .

Si ponga  $v = \phi \in \mathcal{D}(\Omega)$  nella formulazione debole; utilizzando il Teorema 2.19 si ottiene che la *shape derivative*  $u'(\Omega)$  soddisfa

$$\int_{\Omega} \nabla u'(\Omega) \cdot \nabla \phi + \int_{\partial\Omega} \nabla u(\Omega) \cdot \nabla \phi \boldsymbol{\theta} \cdot \mathbf{n} = \int_{\partial\Omega} f \phi \boldsymbol{\theta} \cdot \mathbf{n},$$

da cui, contointegrando per parti il primo termine,

$$-\int_{\Omega} \Delta u'(\Omega) \phi + \int_{\partial\Omega} \frac{\partial u(\Omega)}{\partial \mathbf{n}} \phi + \int_{\partial\Omega} \nabla u(\Omega) \cdot \nabla \phi \boldsymbol{\theta} \cdot \mathbf{n} = \int_{\partial\Omega} f \phi \boldsymbol{\theta} \cdot \mathbf{n}.$$

Essendo  $\phi \in \mathcal{D}(\Omega)$ , si ottiene quindi

$$-\Delta u'(\Omega) = 0, \quad \text{in } \mathcal{D}'(\Omega). \quad (2.27)$$

Si ponga poi  $v = \varphi \in \mathcal{D}(\mathbb{R}^n)$ : dalla condizione al contorno di Dirichlet risulta

$$\int_{\partial\Omega} (u(\Omega) - g)\varphi = 0;$$

derivando questo integrale di bordo, grazie al Teorema 2.19, si ottiene

$$\int_{\partial\Omega} u'(\Omega)\varphi + \int_{\partial\Omega} \left[ \frac{\partial}{\partial \mathbf{n}} \{(u(\Omega) - g)\varphi\} + H(u(\Omega) - g)\varphi \right] \boldsymbol{\theta} \cdot \mathbf{n} = 0,$$

da cui, ricordando che  $u(\Omega) - g = 0$  su  $\partial\Omega$ ,

$$\int_{\partial\Omega} u'(\Omega)\varphi + \int_{\partial\Omega} \frac{\partial}{\partial \mathbf{n}} (u(\Omega) - g)\varphi \boldsymbol{\theta} \cdot \mathbf{n} = 0.$$

Per l'arbitrarietà di  $\varphi$  si ottiene

$$u'(\Omega) = -\frac{\partial}{\partial \mathbf{n}} (u(\Omega) - g) \boldsymbol{\theta} \cdot \mathbf{n}, \quad \text{su } \partial\Omega. \quad (2.28)$$

Dunque la *shape derivative*  $u'(\Omega)$  soddisfa il seguente problema di Poisson:

$$\begin{cases} -\Delta u'(\Omega) = 0, & \text{in } \Omega, \\ u'(\Omega) = -\frac{\partial}{\partial \mathbf{n}} (u(\Omega) - g) \boldsymbol{\theta} \cdot \mathbf{n}, & \text{su } \partial\Omega. \end{cases}$$

## 2.4 Condizioni di ottimalità

Nelle due sezioni seguenti viene mostrato come ricavare le condizioni di ottimalità per un problema di ottimizzazione di forma mediante due differenti approcci che seguono sostanzialmente le idee mostrate in Sezioni 2.1.2 e 2.1.3 nel caso dei problemi di controllo ottimo.

### 2.4.1 L'approccio basato sul problema aggiunto

In questa sezione si mostra come utilizzare il calcolo della *shape derivative*, effettuato in precedenza, per introdurre un opportuno problema aggiunto che permetta di esprimere in modo più conveniente la derivata di forma del funzionale costo.

#### Problema di stato dato da un'equazione Poisson

Di seguito vengono ricavate le condizioni di ottimalità per il problema di ottimizzazione di forma

$$\widehat{\Omega} = \arg \min_{\Omega \in \mathcal{O}_{ad}} J(\Omega),$$

essendo il funzionale costo  $J(\Omega)$  definito da

$$J(\Omega) = \int_{\Omega} |u(\Omega) - \widehat{u}|^2, \quad (2.29)$$

$u(\Omega)$  soluzione dell'equazione di Poisson (2.26),  $\widehat{u} \in L^2(\mathbb{R}^n)$  soluzione di riferimento desiderata.

La derivata rispetto alla forma del funzionale costo (2.29) è data da

$$dJ(\Omega; \boldsymbol{\theta}) = \varphi_1(u'(\Omega)) + \varphi_2(\boldsymbol{\theta} \cdot \mathbf{n}), \quad (2.30)$$



con

$$\begin{aligned}\varphi_1(u'(\Omega)) &= \int_{\Omega} 2(u(\Omega) - \widehat{u})u'(\Omega), \\ \varphi_2(\boldsymbol{\theta} \cdot \mathbf{n}) &= \int_{\partial\Omega} |u(\Omega) - \widehat{u}|^2 \boldsymbol{\theta} \cdot \mathbf{n}.\end{aligned}$$

Considerando il seguente problema aggiunto:

$$\begin{cases} -\Delta p(\Omega) = -2(u(\Omega) - \widehat{u}) & \text{in } \Omega, \\ p(\Omega) = 0 & \text{su } \partial\Omega, \end{cases} \quad (2.31)$$

è possibile esprimere  $dJ(\Omega; \boldsymbol{\theta})$  senza ricorrere all'introduzione della *shape derivative*  $u'(\Omega)$ . Infatti, moltiplicando la (2.31) per  $u'(\Omega)$  e integrando in  $\Omega$  si ottiene, tenendo conto della (2.28),

$$\int_{\Omega} \nabla p(\Omega) \cdot \nabla u'(\Omega) - \int_{\partial\Omega} \left\{ -\frac{\partial}{\partial \mathbf{n}} p(\Omega) \frac{\partial}{\partial \mathbf{n}} (u(\Omega) - g) \boldsymbol{\theta} \cdot \mathbf{n} \right\} = - \int_{\Omega} 2\{u(\Omega) - \widehat{u}\}u'(\Omega).$$

Analogamente, moltiplicando la (2.27) per  $p(\Omega)$  ed integrando su  $\Omega$  si ottiene

$$\int_{\Omega} \nabla u'(\Omega) \cdot \nabla p(\Omega) = 0.$$

Uguagliando le ultime due equazioni si ottiene quindi

$$\int_{\Omega} 2\{u(\Omega) - \widehat{u}\}u'(\Omega) = - \int_{\partial\Omega} \frac{\partial}{\partial \mathbf{n}} p(\Omega) \frac{\partial}{\partial \mathbf{n}} (u(\Omega) - g) \boldsymbol{\theta} \cdot \mathbf{n}$$

da cui, sostituendo in (2.30), si ricava la seguente espressione per la derivata del funzionale costo

$$dJ(\Omega; \boldsymbol{\theta}) = \int_{\partial\Omega} \left\{ |u(\Omega) - \widehat{u}|^2 - \frac{\partial}{\partial \mathbf{n}} p(\Omega) \frac{\partial}{\partial \mathbf{n}} (u(\Omega) - g) \right\} \boldsymbol{\theta} \cdot \mathbf{n}.$$

Tale formula ha senso perché, se  $\Omega$  è di classe  $C^2$ , allora  $u(\Omega), p(\Omega) \in H^2(\Omega)$  per i teoremi di regolarità.

Ricordando infine la seguente relazione:

$$\nabla p(\Omega) = \frac{\partial}{\partial \mathbf{n}} p(\Omega) \mathbf{n},$$

che discende dal fatto che  $p(\Omega)$  e  $u(\Omega) - g$  sono costanti su  $\partial\Omega$ , si può dunque esprimere lo *shape gradient* della *formula di Hadamard* (2.25) come

$$\begin{aligned}g(u(\Omega), p(\Omega)) &= |u(\Omega) - \widehat{u}|^2 - \frac{\partial}{\partial \mathbf{n}} p(\Omega) \frac{\partial}{\partial \mathbf{n}} (u(\Omega) - g) \\ &= |u(\Omega) - \widehat{u}|^2 - \nabla p(\Omega) \cdot \nabla (u(\Omega) - g),\end{aligned}$$

che non coinvolge più la *shape derivative*  $u'(\Omega)$  ed il campo  $\boldsymbol{\theta}$ , ma solo le variabili  $u(\Omega)$  e  $p(\Omega)$ , e può essere quindi calcolato dopo aver risolto il problema di stato ed il problema aggiunto.

In conclusione, il sistema delle condizioni di ottimalità per il problema (2.26)-(2.29) è costituito da

$$\begin{cases} -\Delta u(\Omega) = f, & \text{in } \Omega, \\ u(\Omega) = g, & \text{su } \partial\Omega, \end{cases} \quad \text{problema di stato,}$$

$$\begin{cases} -\Delta p(\Omega) = -2(u(\Omega) - \widehat{u}), & \text{in } \Omega, \\ p(\Omega) = 0, & \text{su } \partial\Omega, \end{cases} \quad \text{problema aggiunto,}$$

$$dJ(\Omega; \boldsymbol{\theta}) = \int_{\partial\Omega} \left\{ |u(\Omega) - \widehat{u}|^2 - \frac{\partial}{\partial \mathbf{n}} p(\Omega) \frac{\partial}{\partial \mathbf{n}} (u(\Omega) - g) \right\} \boldsymbol{\theta} \cdot \mathbf{n}, \quad \text{equazione di sensitività.}$$

### Problema generico di ottimizzazione di forma

Più in generale, si consideri il seguente problema di stato:

$$\text{trovare } u(\Omega) \in \mathcal{V}(\Omega) \text{ tale che } A(\Omega)u(\Omega) = F(\Omega).$$

Utilizzando i risultati forniti nella Sezione 2.3.2, si può ricavare la *shape derivative*  $u'(\Omega)$ , soluzione del seguente problema:

$$\text{trovare } u'(\Omega) \in \mathcal{V}(\Omega) \text{ tale che } A'(\Omega)u'(\Omega) = F'(\Omega). \quad (2.32)$$

essendo<sup>3</sup>  $A'(\Omega) \in \mathcal{L}(\mathcal{V}(\Omega), \mathcal{V}^*(\Omega))$ , spazio degli operatori lineari da  $\mathcal{V}(\Omega)$  a  $\mathcal{V}^*(\Omega)$ , e  $F'(\Omega) = F'(\Omega, \boldsymbol{\theta}) \in \mathcal{V}^*(\Omega)$ .

Derivando rispetto alla forma il generico funzionale costo

$$J(\Omega) = \int_{\Omega} j_{\Omega}(\mathbf{x}, u(\Omega), \nabla u(\Omega)) \, d\mathbf{x} + \int_{\partial\Omega} j_{\partial\Omega}(s, u(\Omega), \nabla u(\Omega)) \, ds,$$

si ottiene, sotto opportune ipotesi di regolarità su  $j_{\Omega}(\mathbf{x}, y, g)$  e  $j_{\partial\Omega}(s, y, g)$  (si veda ad esempio [HM03, HP05])

$$\begin{aligned} dJ(\Omega; \boldsymbol{\theta}) &= \\ &= \left[ \int_{\Omega} \partial_y j_{\Omega}(\mathbf{x}, u(\Omega), \nabla u(\Omega)) u'(\Omega) \, d\mathbf{x} + \int_{\Omega} \nabla_g j_{\Omega}(\mathbf{x}, u(\Omega), \nabla u(\Omega)) \nabla u'(\Omega) \, d\mathbf{x} + \right. \\ &\quad \left. + \int_{\partial\Omega} \partial_y j_{\partial\Omega}(s, u(\Omega), \nabla u(\Omega)) u'(\Omega) \, ds + \int_{\partial\Omega} \nabla_g j_{\partial\Omega}(s, u(\Omega), \nabla u(\Omega)) \nabla u'(\Omega) \, ds \right] + \\ &\quad + \left[ \int_{\partial\Omega} j_{\Omega}(s, u(\Omega), \nabla u(\Omega)) \boldsymbol{\theta} \cdot \mathbf{n} \, ds + \right. \\ &\quad \left. + \int_{\partial\Omega} \left\{ \frac{\partial}{\partial \mathbf{n}} j_{\partial\Omega}(s, u(\Omega), \nabla u(\Omega)) + H j_{\partial\Omega}(s, u(\Omega), \nabla u(\Omega)) \right\} \boldsymbol{\theta} \cdot \mathbf{n} \, ds \right] \\ &= \varphi_1(u'(\Omega)) + \varphi_2(\boldsymbol{\theta} \cdot \mathbf{n}), \end{aligned}$$

avendo posto

$$\begin{aligned} \varphi_1(u'(\Omega)) &= \int_{\Omega} \partial_y j_{\Omega}(\mathbf{x}, u(\Omega), \nabla u(\Omega)) u'(\Omega) \, d\mathbf{x} + \int_{\Omega} \nabla_g j_{\Omega}(\mathbf{x}, u(\Omega), \nabla u(\Omega)) \nabla u'(\Omega) \, d\mathbf{x} + \\ &\quad + \int_{\partial\Omega} \partial_y j_{\partial\Omega}(s, u(\Omega), \nabla u(\Omega)) u'(\Omega) \, ds + \int_{\partial\Omega} \nabla_g j_{\partial\Omega}(s, u(\Omega), \nabla u(\Omega)) \nabla u'(\Omega) \, ds \\ \varphi_2(\boldsymbol{\theta} \cdot \mathbf{n}) &= \int_{\partial\Omega} j_{\Omega}(s, u(\Omega), \nabla u(\Omega)) \boldsymbol{\theta} \cdot \mathbf{n} \, ds + \\ &\quad + \int_{\partial\Omega} \left\{ \frac{\partial}{\partial \mathbf{n}} j_{\partial\Omega}(s, u(\Omega), \nabla u(\Omega)) + H j_{\partial\Omega}(s, u(\Omega), \nabla u(\Omega)) \right\} \boldsymbol{\theta} \cdot \mathbf{n} \, ds. \end{aligned}$$

Indicando con  $B^*$  l'aggiunto di un generico operatore  $B$ , si definisce lo stato aggiunto  $p(\Omega)$ , soluzione di

$$\text{trovare } p(\Omega) \in \mathcal{V}(\Omega) \text{ tale che } (A')(\Omega)^* p(\Omega) = \varphi_1;$$

in questo modo è possibile ottenere

$$\begin{aligned} \varphi_1(u'(\Omega)) &= \langle \varphi_1, u'(\Omega) \rangle_{\mathcal{V}^*(\Omega) \times \mathcal{V}(\Omega)} = \langle (A')(\Omega)^* p, u'(\Omega) \rangle_{\mathcal{V}^*(\Omega) \times \mathcal{V}(\Omega)} = \\ &= \langle A'(\Omega) u'(\Omega), p(\Omega) \rangle_{\mathcal{V}^*(\Omega) \times \mathcal{V}(\Omega)} = \langle F'(\Omega, \boldsymbol{\theta}), p(\Omega) \rangle_{\mathcal{V}^*(\Omega) \times \mathcal{V}(\Omega)}. \end{aligned}$$

<sup>3</sup>Per evitare conflitti di notazione si mantiene l'apice per le quantità collegate alla *shape derivative* e si cambia il simbolo degli spazi duali ed operatori aggiunti in un asterisco.

La seconda uguaglianza è giustificata dalla definizione di  $\varphi_1$ , la terza dall'*identità di Lagrange* e la quarta dalla (2.32).

Questo permette di semplificare la formulazione della derivata euleriana di  $J$ , eliminando la dipendenza da  $u'(\Omega)$ : infatti, sostituendo l'equazione precedente si ottiene

$$dJ(\Omega; \boldsymbol{\theta}) = \langle F'(\Omega, \boldsymbol{\theta}), p(\Omega) \rangle_{\mathcal{V}^*(\Omega) \times \mathcal{V}(\Omega)} + \varphi_2(\boldsymbol{\theta} \cdot \mathbf{n}),$$

da cui si ricava il seguente sistema di condizioni di ottimalità

$$\begin{cases} \text{trovare } u(\Omega) \in \mathcal{V}(\Omega) \text{ tale che } Au(\Omega) = F, & \text{problema di stato,} \\ \text{trovare } p(\Omega) \in \mathcal{V}(\Omega) \text{ tale che } (A')^*p(\Omega) = \varphi_1, & \text{problema aggiunto,} \\ dJ(\Omega; \boldsymbol{\theta}) = \langle F'(\boldsymbol{\theta}), p(\Omega) \rangle_{\mathcal{V}^*(\Omega) \times \mathcal{V}(\Omega)} + \varphi_2(\boldsymbol{\theta} \cdot \mathbf{n}), & \text{equazione di sensitività.} \end{cases}$$

### 2.4.2 Metodo dei moltiplicatori di Lagrange

Come visto nella sezione precedente, il calcolo della derivata rispetto alla forma di un funzionale costo necessita della scrittura della *shape derivative* del problema di stato, sebbene essa non compaia nel risultato finale. Esiste una strategia più rapida per la derivazione di un funzionale costo e per la scrittura del problema aggiunto, basata sul metodo dei moltiplicatori di Lagrange – introdotto in quest'ambito da C ea in [C ea86] – che verr a presentato per mezzo di due esempi tratti da [All07].

#### Problema di stato dato da un'equazione Poisson con condizioni al bordo di Neumann

Si consideri il problema di ottimizzazione di forma

$$\widehat{\Omega} = \arg \min_{\Omega \in \mathcal{O}_{ad}} J(\Omega),$$

essendo il funzionale costo  $J(\Omega)$  dato da

$$J(\Omega) = \int_{\Omega} j(u(\Omega)),$$

e  $u(\Omega)$  la soluzione della seguente equazione di Poisson, con condizioni al bordo di Neumann,

$$\begin{cases} -\Delta u(\Omega) = f, & \text{in } \Omega, \\ \frac{\partial u(\Omega)}{\partial \mathbf{n}} = h, & \text{su } \partial\Omega. \end{cases}$$

Sommando il funzionale costo alla formulazione debole dell'equazione di stato si ottiene il seguente funzionale Lagrangiano

$$\mathcal{L}(\Omega, v; q) = \int_{\Omega} j(v) + \int_{\Omega} \nabla v \cdot \nabla q - f q + \int_{\partial\Omega} h q,$$

con  $\Omega \subset \mathbb{R}^n$ ,  $v \in H^1(\mathbb{R}^n)$ ,  $q \in H^1(\mathbb{R}^n)$ . È importante notare che lo spazio  $H^1(\mathbb{R}^n)$  non dipende da  $\Omega$ , e quindi le tre variabili  $\Omega$ ,  $v$  e  $q$  sono indipendenti; la variabile  $q$  assume inoltre il ruolo di moltiplicatore di Lagrange. Procedendo come nella Sezione 2.1.3, occorre esplicitare le derivate di  $\mathcal{L}$  rispetto alle tre variabili:

- o la derivata parziale di  $\mathcal{L}$  rispetto a  $q$  in direzione  $\phi \in H^1(\mathbb{R}^n)$  è

$$\left\langle \frac{\partial \mathcal{L}}{\partial q}(\Omega, v; q), \phi \right\rangle = \int_{\Omega} \nabla v \cdot \nabla \phi - f \phi + \int_{\partial\Omega} h \phi,$$

e richiedendo che all'ottimo essa si annulli  $\forall \phi \in H^1(\mathbb{R}^n)$ , si ottiene la formulazione debole dell'equazione di stato.

o la derivata parziale di  $\mathcal{L}$  rispetto a  $v$  in direzione  $\phi \in H^1(\mathbb{R}^n)$  è

$$\left\langle \frac{\partial \mathcal{L}}{\partial v}(\Omega, v; q), \phi \right\rangle = \int_{\Omega} j'(v)\phi + \int_{\Omega} \nabla \phi \nabla q,$$

da cui, eseguendo una integrazione per parti,

$$\left\langle \frac{\partial \mathcal{L}}{\partial v}(\Omega, v; q), \phi \right\rangle = \int_{\Omega} j'(v)\phi - \int_{\Omega} \phi \Delta q + \int_{\partial \Omega} \phi \frac{\partial q}{\partial \mathbf{n}}.$$

Richiedendo che all'ottimo essa si annulli  $\forall \phi \in H^1(\mathbb{R}^n)$ , si ottiene, rispettivamente:

1. l'equazione aggiunta, se  $\phi$  è a supporto compatto in  $\Omega$ :

$$-\Delta p = -j'(u), \quad \text{in } \Omega;$$

2. le condizioni al bordo dell'equazione aggiunta, se  $\phi$  è a supporto compatto in  $\partial \Omega$ :

$$\frac{\partial p}{\partial \mathbf{n}} = 0, \quad \text{su } \partial \Omega.$$

Si ricava quindi che lo stato aggiunto è soluzione del seguente problema:

$$\begin{cases} -\Delta p(\Omega) = -j'(u(\Omega)), & \text{in } \Omega, \\ \frac{\partial p(\Omega)}{\partial \mathbf{n}} = 0, & \text{su } \partial \Omega; \end{cases}$$

o la derivata parziale di  $\mathcal{L}$  rispetto a  $\Omega$  in direzione  $\boldsymbol{\theta} \in W^{1,\infty}(\mathbb{R}^n; \mathbb{R}^n)$  è

$$\frac{\partial \mathcal{L}}{\partial \Omega}(\Omega, v; q)(\boldsymbol{\theta}) = \int_{\partial \Omega} \left[ j'(v) + \nabla v \cdot \nabla q - fq - \frac{\partial(hq)}{\partial \mathbf{n}} - Hgq \right] \boldsymbol{\theta} \cdot \mathbf{n}.$$

Si può mostrare infine che, all'ottimo,

$$dJ(\widehat{\Omega}; \boldsymbol{\theta}) = \frac{\partial \mathcal{L}}{\partial \Omega}(\widehat{\Omega}, u(\widehat{\Omega}); p(\widehat{\Omega}))(\boldsymbol{\theta}).$$

Infatti, per ogni  $\Omega \subset \mathbb{R}^n$  e  $q \in H^1(\mathbb{R}^n)$ , vale la seguente relazione

$$J(\Omega) = \mathcal{L}(\Omega, u(\Omega); q),$$

dal momento che  $u(\Omega)$  soddisfa l'equazione di stato in  $\Omega$ ; derivando formalmente la precedente relazione rispetto ad  $\Omega$  si ottiene, tramite la regola di derivazione di una funzione composta

$$dJ(\Omega; \boldsymbol{\theta}) = \frac{\partial \mathcal{L}}{\partial \Omega}(\Omega, u(\Omega); q)(\boldsymbol{\theta}) + \left\langle \frac{\partial \mathcal{L}}{\partial v}(\Omega, u(\Omega); q), u'(\Omega) \right\rangle,$$

da cui la tesi, in quanto l'ultimo termine si annulla se  $\Omega = \widehat{\Omega}$  e  $q = p(\widehat{\Omega})$  grazie alla definizione del problema aggiunto. Tuttavia, a causa di quest'ultimo passaggio, il risultato è solo formale: infatti non è noto a priori se sia legittimo derivare  $u(\Omega)$  rispetto al dominio ed applicare la formula della derivata composta.

**Problema di stato dato da un'equazione Poisson con condizioni al bordo di Dirichlet**

Si consideri ora il problema di ottimizzazione di forma

$$\widehat{\Omega} = \arg \min_{\Omega \in \mathcal{O}_{ad}} J(\Omega),$$

essendo il funzionale costo  $J(\Omega)$  dato da

$$J(\Omega) = \int_{\Omega} j(u(\Omega)),$$

e  $u(\Omega)$  soluzione della seguente equazione di Poisson, con condizioni al bordo di Dirichlet,

$$\begin{cases} -\Delta u(\Omega) = f, & \text{in } \Omega, \\ u(\Omega) = g, & \text{su } \partial\Omega. \end{cases}$$

È ancora possibile, come nel caso delle condizioni di Neumann, definire il funzionale Lagrangiano  $\mathcal{L}(\Omega, v; q)$  sommando il funzionale costo e la formulazione debole del problema di stato. Tuttavia, affinché  $v$  rispetti la condizione di Dirichlet  $v = g$  su  $\partial\Omega$ , è necessario prescriverne la traccia su  $\partial\Omega$ : ma in questo modo le variabili  $\Omega$  ed  $v$  non risulterebbero più indipendenti. È necessario quindi introdurre un secondo moltiplicatore di Lagrange  $\lambda$  per garantire la condizione al bordo, ottenendo il seguente funzionale Lagrangiano:

$$\mathcal{L}(\Omega, v; q, \lambda) = \int_{\Omega} j(v) + \int_{\Omega} (-\Delta v - f)q + \int_{\partial\Omega} \lambda(v - g),$$

con  $\Omega \subset \mathbb{R}^n$ ,  $v \in H^1(\mathbb{R}^n)$ ,  $q \in H^1(\mathbb{R}^n)$ ,  $\lambda \in H^1(\mathbb{R}^n)$ , ripristinando in questo modo l'indipendenza tra le variabili  $\Omega, v, q$  e  $\lambda$ . Procedendo come nella Sezione 2.1.3, occorre esplicitare le derivate di  $\mathcal{L}$  rispetto alle quattro variabili:

- la derivata parziale di  $\mathcal{L}$  rispetto a  $q$  in direzione  $\phi \in H^1(\mathbb{R}^n)$  è

$$\left\langle \frac{\partial \mathcal{L}}{\partial q}(\Omega, v; q, \lambda), \phi \right\rangle = \int_{\Omega} (-\Delta v - f)\phi,$$

e richiedendo che all'ottimo essa si annulli  $\forall \phi \in H^1(\mathbb{R}^n)$ , si ottiene l'equazione di stato.

- la derivata parziale di  $\mathcal{L}$  rispetto a  $\lambda$  in direzione  $\varphi \in H^1(\mathbb{R}^n)$  è

$$\left\langle \frac{\partial \mathcal{L}}{\partial \lambda}(\Omega, v; q, \lambda), \varphi \right\rangle = \int_{\partial\Omega} \varphi(v - g),$$

e richiedendo che all'ottimo essa si annulli  $\forall \varphi \in H^1(\mathbb{R}^n)$ , si ottengono le condizioni al bordo dell'equazione di stato.

- la derivata parziale di  $\mathcal{L}$  rispetto a  $v$  in direzione  $\phi \in H^1(\mathbb{R}^n)$  è

$$\left\langle \frac{\partial \mathcal{L}}{\partial v}(\Omega, v; q, \lambda), \phi \right\rangle = \int_{\Omega} j'(v)\phi - \int_{\Omega} q\Delta\phi + \int_{\partial\Omega} \lambda\phi,$$

da cui, integrando per parti, si ricava

$$\left\langle \frac{\partial \mathcal{L}}{\partial v}(\Omega, v; q, \lambda), \phi \right\rangle = \int_{\Omega} (j'(v) - \Delta q)\phi + \int_{\partial\Omega} \phi \left( \lambda + \frac{\partial q}{\partial n} \right) - \int_{\partial\Omega} \frac{\partial \phi}{\partial n} q.$$

Richiedendo che all'ottimo essa si annulli  $\forall \phi \in H^1(\mathbb{R}^n)$ , si ottiene, rispettivamente:

1. l'equazione aggiunta, se  $\phi$  è a supporto compatto in  $\Omega$ :

$$-\Delta p = -j'(u), \quad \text{in } \Omega;$$

2. le condizioni al bordo dell'equazione aggiunta, se  $\phi$  è a supporto compatto in  $\partial\Omega$ :

(a) se  $\phi$  è nulla su  $\partial\Omega$ , allora si ricavano le condizioni al bordo

$$p = 0, \quad \text{su } \partial\Omega;$$

(b) se  $\phi$  è qualsiasi su  $\partial\Omega$ , allora si ricava anche

$$\lambda = -\frac{\partial p}{\partial n}.$$

Si ottiene quindi che lo stato aggiunto è soluzione del seguente problema:

$$\begin{cases} -\Delta p(\Omega) = -j'(u(\Omega)), & \text{in } \Omega, \\ p(\Omega) = 0, & \text{su } \partial\Omega; \end{cases}$$

o la derivata parziale di  $\mathcal{L}$  rispetto a  $\Omega$  in direzione  $\boldsymbol{\theta} \in W^{1,\infty}(\mathbb{R}^n; \mathbb{R}^n)$ , valutata all'ottimo<sup>4</sup>, è

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \Omega}(\widehat{\Omega}, u(\widehat{\Omega}); p(\widehat{\Omega}), \lambda(\widehat{\Omega}))(\boldsymbol{\theta}) &= dJ(\widehat{\Omega}; \boldsymbol{\theta}) = \int_{\partial\widehat{\Omega}} j(u(\widehat{\Omega})) \boldsymbol{\theta} \cdot \mathbf{n} + \\ &+ \int_{\partial\widehat{\Omega}} (-\Delta u(\widehat{\Omega}) - f)p(\widehat{\Omega}) \boldsymbol{\theta} \cdot \mathbf{n} + \int_{\partial\widehat{\Omega}} \frac{\partial\{\lambda(\widehat{\Omega})(u(\widehat{\Omega}) - g)\}}{\partial n} \boldsymbol{\theta} \cdot \mathbf{n} + \\ &+ \int_{\partial\widehat{\Omega}} H\lambda(\widehat{\Omega})(u(\widehat{\Omega}) - g) \boldsymbol{\theta} \cdot \mathbf{n}. \end{aligned}$$

Le condizioni al contorno di  $u(\widehat{\Omega})$  e  $p(\widehat{\Omega})$  rendono nulli rispettivamente il quarto e secondo integrale. Sostituendo l'espressione di  $\lambda(\widehat{\Omega})$  si ottiene infine

$$dJ(\widehat{\Omega}; \boldsymbol{\theta}) = \int_{\partial\widehat{\Omega}} \left\{ j(u(\widehat{\Omega})) - \frac{\partial}{\partial n} p(\widehat{\Omega}) \frac{\partial}{\partial n} (u(\widehat{\Omega}) - g) \right\} \boldsymbol{\theta} \cdot \mathbf{n}.$$

Si noti che, se  $j(v) = |v - \widehat{u}|^2$ ,  $\widehat{u} \in L^2(\mathbb{R}^n)$ , si ottengono le medesime condizioni di ottimalità della Sezione 2.4.1.

### Problema generico di ottimizzazione di forma

Il metodo dei moltiplicatori di Lagrange può essere applicato anche ad un generico problema di ottimizzazione di forma, seguendo le stesse idee dei due esempi precedenti: si veda ad esempio [Céa86]. In particolare, in questo elaborato, il metodo dei moltiplicatori verrà nuovamente applicato nella Sezione 3.3.4 per problemi di ottimizzazione di forma con sistema fisico governato dalle equazioni di Stokes stazionarie e non stazionarie.

<sup>4</sup>L'uguaglianza  $\frac{\partial \mathcal{L}}{\partial \Omega}(\widehat{\Omega}, u(\widehat{\Omega}); p(\widehat{\Omega}), \lambda(\widehat{\Omega}))(\boldsymbol{\theta}) = dJ(\widehat{\Omega}; \boldsymbol{\theta})$  si ricava con le stesse considerazioni formali illustrate per il caso con condizioni di Neumann.

## Capitolo 3

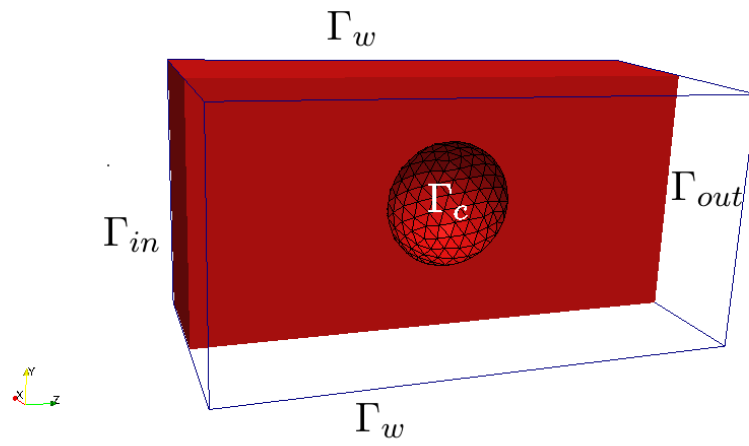
# Ottimizzazione di forma per problemi di fluidodinamica

In questo capitolo vengono presentati alcuni problemi di ottimizzazione di forma in ambito fluidodinamico ed in particolare per fluidi viscosi incompressibili governati dalle equazioni di Stokes, stazionarie e non stazionarie. Rispetto alla formulazione astratta presentata nel capitolo precedente, le variabili di stato caratterizzano la velocità  $\mathbf{u}$  e pressione  $p$  del fluido, mentre alcuni tipici funzionali costo misurano l'energia dissipata, la resistenza (o *drag*) o la vorticità. Vengono esaminati sia flussi esterni, sia flussi interni: per la prima categoria la geometria di riferimento è quella di un corpo immerso, individuato dalla frontiera  $\Gamma_c$  (Figura 3.1(a)-(b)); per quanto riguarda i flussi interni si considera invece la geometria di un bypass aorto-coronarico (Figura 3.1(c)).

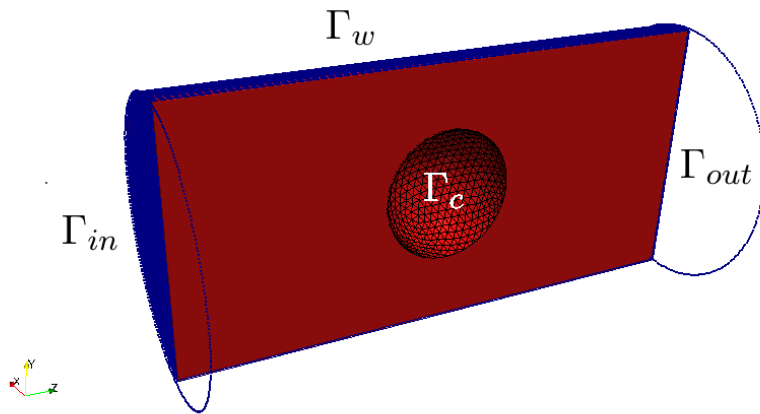
Nella Sezione 3.1 viene brevemente richiamata la modellazione matematica di un fluido viscoso incompressibile, le equazioni di Navier-Stokes e la loro chiusura con opportune condizioni al contorno.

La Sezione 3.2 è dedicata a problemi di ottimizzazione di forma con equazione di stato di Stokes stazionario e minimizzazione dell'energia dissipata: dopo aver ricordato la formulazione debole del problema di stato e la sua buona posizione (Sezione 3.2.1), nella Sezione 3.2.2 viene dimostrata l'esistenza di almeno una forma ottima tramite l'applicazione del Teorema 2.4 di esistenza; la Sezione 3.2.3 conclude questa analisi mostrando anche un risultato di simmetria della forma ottima. Per ricavare il sistema delle condizioni di ottimalità, sarà necessario anzitutto ricavare la *shape derivative* del problema di stato, come mostrato nella Sezione 3.2.4, ed introdurre un opportuno problema aggiunto (Sezione 3.2.5). Si mostrerà infine nella Sezione 3.2.6 che, se sia il termine forzante sia le condizioni sul bordo di *outflow* sono nulli, allora è possibile risolvere esplicitamente il problema aggiunto, semplificando dunque le condizioni di ottimalità, con un notevole risparmio nella successiva risoluzione numerica approssimata.

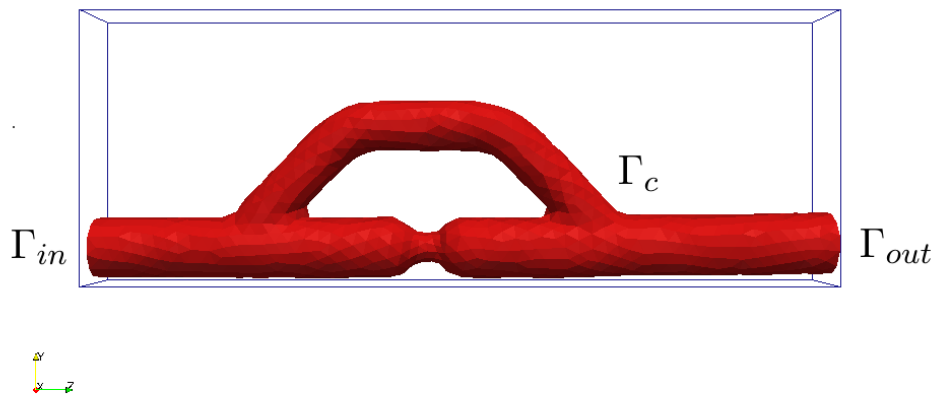
Questa struttura sarà in buona parte ripercorsa nella Sezione 3.3, in cui il sistema fisico è governato dal problema di Stokes non stazionario: la formulazione debole sarà introdotta nella Sezione 3.3.1, l'esistenza di una forma ottima sarà discussa nella Sezione 3.3.2. Per quanto riguarda la scrittura delle condizioni di ottimalità, dopo aver ricavato la *shape derivative* del problema di stato nella Sezione 3.3.3, nella Sezione 3.3.4 si procederà all'introduzione di un opportuno problema aggiunto, considerando due diverse possibilità per quanto riguarda la condizione iniziale del problema di stato non stazionario.



(a) Corpo immerso.



(b) Corpo immerso (2).



(c) Bypass aorto-coronarico.

Figura 3.1: Geometrie.



### 3.1 Descrizione del modello fluidodinamico

In questa sezione vengono brevemente richiamate le equazioni di Navier-Stokes per un fluido viscoso incompressibile.

Sia  $\rho = \rho(\mathbf{x}, t)$  la densità del fluido ed  $\mathbf{u} = \mathbf{u}(\mathbf{x}, t)$  la sua velocità, con  $\mathbf{x} \in \mathbb{R}^3$  e  $t > 0$ : la conservazione della massa, in un arbitrario volume occupato dal fluido, permette di scrivere la seguente *equazione di continuità*:

$$\frac{\partial \rho}{\partial t} + \operatorname{div} \rho \mathbf{u} = 0;$$

la conservazione della quantità di moto, in un arbitrario volume occupato dal fluido, si traduce invece nella seguente equazione:

$$\rho \frac{\partial \mathbf{u}}{\partial t} + \rho(\mathbf{u} \cdot \nabla) \mathbf{u} - \operatorname{div} T = \rho \mathbf{f},$$

dove  $\mathbf{f}$  rappresenta le forze esterne agenti sul sistema (per unità di massa) e  $T$  è il tensore degli sforzi di Cauchy, per il quale è necessario imporre una *legge costitutiva* che tenga conto del comportamento meccanico del fluido in esame. In particolare verrà assunto il modello di *fluido Newtoniano*, in base alla quale il tensore degli sforzi è una funzione lineare delle derivate di  $\mathbf{u}$ , e dipende da una funzione scalare  $p = p(\mathbf{x}, t)$ , detta pressione, e dalla viscosità dinamica  $\mu$  del fluido:

$$T = T(\mathbf{u}, p) = -pI + \mu(\nabla \mathbf{u} + \nabla^T \mathbf{u}).$$

L'introduzione del tensore velocità di deformazione, dato da

$$D(\mathbf{u}) = \frac{1}{2}(\nabla \mathbf{u} + \nabla^T \mathbf{u}),$$

permette di esprimere il tensore degli sforzi come somma di una componente normale dovuta alla pressione e di una componente di sforzo viscoso

$$T = T(\mathbf{u}, p) = -pI + 2\mu D(\mathbf{u}).$$

L'equazione di conservazione della quantità di moto può essere dunque riscritta come

$$\rho \frac{\partial \mathbf{u}}{\partial t} + \rho(\mathbf{u} \cdot \nabla) \mathbf{u} + \nabla p - 2 \operatorname{div} \{ \mu D(\mathbf{u}) \} = \rho \mathbf{f}.$$

Ipotizzando infine che le proprietà fisiche  $\rho$  e  $\mu$  del fluido siano costanti, si ottiene il seguente problema di *Navier-Stokes* non stazionario:

$$\begin{cases} \frac{\partial \mathbf{u}}{\partial t} - \nu \Delta \mathbf{u} + (\mathbf{u} \cdot \nabla) \mathbf{u} + \nabla p = \mathbf{f}, & \text{in } \Omega \times (0, \tau), \\ \operatorname{div} \mathbf{u} = 0, & \text{in } \Omega \times (0, \tau), \end{cases}$$

avendo indicato con  $\nu = \mu/\rho$  la viscosità cinematica e riscaldato la pressione con la densità (mantendo però lo stesso simbolo  $p$ ).

La descrizione del moto del fluido viene poi completata imponendo una condizione iniziale sulla velocità ed opportune condizioni al contorno: una scelta frequente prevede di assegnare sulla frontiera  $\Gamma_{in}$  di inflow una condizione di Dirichlet sulla velocità  $\mathbf{u} = \mathbf{u}_{in}$ , e sulla frontiera  $\Gamma_{out}$  una condizione di Neumann, che coinvolge lo sforzo totale. La restante parte della frontiera verrà considerata come una parete fissa: si impone dunque  $\mathbf{u} = \mathbf{0}$  su  $\Gamma_0$ , indicando con  $\Gamma_0 = \Gamma_c \cup \Gamma_w$  nel caso della geometria di Figura 3.1(a)-(b) e  $\Gamma_0 = \Gamma_c$  nel caso della geometria di Figura 3.1(c). Si denomina infine  $\Gamma_D$  la porzione di frontiera sulla quale sono imposte condizioni di Dirichlet, cioè  $\Gamma_D = \Gamma_{in} \cup \Gamma_0$ .

Nel seguito verrà considerato il modello semplificato ottenuto trascurando il termine convettivo (problema di *Stokes* non stazionario)

$$\begin{cases} \frac{\partial \mathbf{u}}{\partial t} - \nu \Delta \mathbf{u} + \nabla p = \mathbf{f}, & \text{in } \Omega \times (0, \tau), \\ \operatorname{div} \mathbf{u} = 0, & \text{in } \Omega, \\ \text{condizioni al contorno,} & \text{su } \partial\Omega \times (0, \tau), \\ \text{condizione iniziale,} & \text{su } \Omega \times \{0\}; \end{cases}$$

questo modello verrà usato come problema di stato (eventualmente trascurando anche la dipendenza temporale) nei problemi di ottimizzazione di forma considerati nelle prossime sezioni.

Rimane infine da specificare il criterio di ottimizzazione: nel caso dell'ottimizzazione di un profilo immerso in flusso come in Figura 3.1(a)-(b), una scelta naturale consiste nella minimizzazione della resistenza (o *drag*):

$$- \int_0^T \int_{\Gamma_c} \hat{\mathbf{u}}_{in} \cdot T(\mathbf{u}, p) \mathbf{n} \, ds \, dt,$$

essendo  $\hat{\mathbf{u}}_{in}$  il versore diretto come  $\mathbf{u}_{in}$ ; una scelta analoga prevede di considerare la minimizzazione dell'energia dissipata

$$\frac{\nu}{2} \int_0^T \int_{\Omega} |\nabla \mathbf{u}|^2 \, d\mathbf{x} \, dt.$$

Nel caso di Figura 3.1(c), due funzionali costo molto studiati (si veda ad esempio [PLN<sup>+</sup>10, Roz05a, Roz05b]) sono l'energia dissipata ed il seguente funzionale costo, dipendente dalla vorticità,

$$\frac{1}{2} \int_0^T \int_{\Omega} |\nabla \times \mathbf{u}|^2 \, d\mathbf{x} \, dt.$$

Per semplicità di esposizione si considera per tutti i casi di Figura 3.1 lo stesso funzionale costo, dato dall'energia dissipata. Questa scelta è anche parzialmente giustificata dal fatto che, a meno di considerare condizioni di Dirichlet anche su  $\Gamma_{out}$  e limitatamente al caso stazionario, si può mostrare l'equivalenza tra il *drag* e l'energia dissipata per la configurazione del corpo immerso (si veda ad esempio [Man08, Ric95, BFCLS97]); per quanto riguarda invece la configurazione del bypass aorto-coronarico, la minimizzazione dell'energia dissipata e della vorticità conducono allo stesso termine forzante nel problema aggiunto (si veda ad esempio [Roz05b]).

## 3.2 Problema di Stokes stazionario

Dati  $\mathbf{f} \in \{[H^1(B)]^3\}'$ ,  $\mathbf{u}_{in} \in [H^{1/2}(\Gamma_{in})]^3$  e  $\mathbf{h} \in [H^{-1/2}(\Gamma_{out})]^3$ , si consideri il seguente problema di Stokes stazionario<sup>1</sup>:

$$\begin{cases} -\nu \Delta \mathbf{u}(\Omega) + \nabla p(\Omega) = \mathbf{f}, & \text{in } \Omega, \\ \operatorname{div} \mathbf{u}(\Omega) = 0, & \text{in } \Omega, \\ \mathbf{u}(\Omega) = \mathbf{u}_{in}, & \text{su } \Gamma_{in}, \\ \mathbf{u}(\Omega) = \mathbf{0}, & \text{su } \Gamma_0, \\ \nu \frac{\partial \mathbf{u}(\Omega)}{\partial \mathbf{n}} - p(\Omega) \mathbf{n} = \mathbf{h}, & \text{su } \Gamma_{out}, \end{cases} \quad (\mathcal{S}_{steady})$$

ed il seguente funzionale costo

$$J(\Omega) = \frac{\nu}{2} \int_{\Omega} |\nabla \mathbf{u}(\Omega)|^2 \, d\mathbf{x}. \quad (\mathcal{J}_{steady})$$

<sup>1</sup> $\Omega$  indica una qualsiasi delle geometrie di Figura 3.1. La notazione  $\Omega^{(i)}$ ,  $i = a, b, c$ , sarà utilizzata, quando necessario, per riferirsi solo al dominio in Figura 3.1( $i$ ),  $i = a, b, c$ .

Nelle seguenti sezioni verrà fornita l'analisi di esistenza e la scrittura delle condizioni di ottimalità per il problema di ottimizzazione di forma

$$\widehat{\Omega} = \arg \min_{\Omega \in \mathcal{O}_{ad}} J(\Omega). \quad (3.1)$$

### 3.2.1 Formulazione debole del problema di stato

In questa sezione viene richiamata la formulazione debole del problema di Stokes stazionario e la sua buona posizione; indicando con

$$\begin{aligned} Q(\Omega) &= \begin{cases} L^2(\Omega), & \text{se } \Gamma_{out} \neq \emptyset, \\ L^2(\Omega) \setminus \mathbb{R}, & \text{se } \Gamma_{out} = \emptyset, \end{cases} & V(\Omega) &= [H^1(\Omega)]^3, \\ V_D(\Omega) &= [H_{\Gamma_0}^1(\Omega)]^3 \cap \{\mathbf{v} \in V(\Omega) : \mathbf{v}|_{\Gamma_D} = \mathbf{u}_D\}, & V_0(\Omega) &= [H_{\Gamma_D}^1(\Omega)]^3, \end{aligned}$$

e con  $\mathbf{u}_D \in H^{1/2}(\Gamma_D)$  la funzione definita come

$$\mathbf{u}_D = \begin{cases} \mathbf{u}_{in}, & \text{su } \Gamma_{in}, \\ \mathbf{0}, & \text{su } \Gamma_0, \end{cases}$$

la formulazione debole di ( $S_{steady}$ ) è:

$$\begin{aligned} \text{trovare } \mathbf{u} \in V_D(\Omega), p \in Q(\Omega) : \\ a(\mathbf{u}, \mathbf{v}) + b(\mathbf{v}, p) &= F(\mathbf{v}), & \forall \mathbf{v} \in V_0(\Omega), \\ b(\mathbf{u}, q) &= \mathbf{0}, & \forall q \in Q(\Omega), \end{aligned} \quad (3.2)$$

dove

$$\begin{aligned} a : V(\Omega) \times V(\Omega) &\rightarrow \mathbb{R}, & a(\mathbf{u}, \mathbf{v}) &= \int_{\Omega} \nu \nabla \mathbf{u} : \nabla \mathbf{v}, \\ b : V(\Omega) \times Q(\Omega) &\rightarrow \mathbb{R}, & b(\mathbf{v}, p) &= - \int_{\Omega} p \operatorname{div} \mathbf{v}, \\ F : V(\Omega) &\rightarrow \mathbb{R}, & F(\mathbf{v}) &= \int_{\Omega} \mathbf{f} \cdot \mathbf{v} + \int_{\Gamma_{out}} \mathbf{h} \cdot \mathbf{v}. \end{aligned}$$

La continuità delle forme bilineari  $a(\mathbf{u}, \mathbf{v})$  e  $b(\mathbf{v}, p)$  è immediata conseguenza della disuguaglianza di Cauchy-Schwarz; si indica con  $\gamma = \nu$  la costante di continuità  $a(\mathbf{u}, \mathbf{v})$  e con  $\delta = \sqrt{n}$  la costante di continuità di  $b(\mathbf{v}, p)$ . La forma bilineare  $a(\mathbf{u}, \mathbf{v})$  è coerciva su  $V(\Omega) \times V(\Omega)$ , con costante di coercività  $\alpha = \frac{\nu}{1+C_P^2}$ , grazie alla disuguaglianza di Poincaré. Inoltre la forma  $b(\mathbf{v}, p)$  verifica la seguente condizione *inf-sup*:

$$\exists \beta > 0 : \quad \inf_{q \in Q(\Omega)} \sup_{\mathbf{v} \in V_0(\Omega)} \frac{b(\mathbf{v}, q)}{\|\mathbf{v}\|_{V_0(\Omega)} \|q\|_{Q(\Omega)}} \geq \beta.$$

L'esistenza di un'unica soluzione  $\mathbf{u}$  è quindi conseguenza del lemma di Lax-Milgram applicato al sottospazio  $V_{\operatorname{div}}(\Omega) = \{\mathbf{v} \in V_0(\Omega) : \operatorname{div} \mathbf{v} = 0\}$ . Inoltre, è possibile ricavare una *disuguaglianza di energia* per la velocità: infatti, supponendo per semplicità  $\mathbf{u}_D = \mathbf{0}$ , scegliendo  $\mathbf{v} = \mathbf{u}$  nella (3.2) si ottiene

$$\alpha \|\mathbf{u}\|_{V_D(\Omega)}^2 \leq a(\mathbf{u}, \mathbf{u}) = F(\mathbf{u}) \leq \left( \|\mathbf{f}\|_{V'(\Omega)} + C_T \|\mathbf{h}\|_{H^{-1/2}(\Gamma_{out})} \right) \|\mathbf{u}\|_{V_D(\Omega)},$$

essendo  $C_T$  la costante di traccia. Si ricava quindi

$$\|\mathbf{u}\|_{V_D(\Omega)} \leq \frac{1}{\alpha} \left( \|\mathbf{f}\|_{V'(\Omega)} + C_T \|\mathbf{h}\|_{H^{-1/2}(\Gamma_{out})} \right),$$

che è la stima cercata per la velocità  $\mathbf{u}$ . La dimostrazione di esistenza ed unicità della pressione  $p$  è conseguenza del seguente lemma:

**Lemma 3.1 ([QV08]).** *Sia  $\Omega$  un dominio limitato in  $\mathbb{R}^n$ , di frontiera lipschitziana. Sia  $L \in V'(\Omega)$ . Allora*

$$L(\mathbf{v}) = 0, \quad \forall \mathbf{v} \in V_{\text{div}}(\Omega)$$

*se e solo esiste un'unica funzione  $p \in Q(\Omega)$  tale che*

$$L(\mathbf{v}) = b(\mathbf{v}, p), \quad \forall \mathbf{v} \in V(\Omega).$$

Per dimostrare l'esistenza e l'unicità della pressione basta quindi scegliere

$$L(\mathbf{v}) = F(\mathbf{v}) - a(\mathbf{u}, \mathbf{v}) \in V'(\Omega).$$

La stima di stabilità per la pressione  $p$  è ricavata utilizzando la condizione *inf-sup*; risulta infatti:

$$\begin{aligned} \|p\|_{Q(\Omega)} &\leq \frac{1}{\beta} \sup_{\mathbf{v} \in V_0(\Omega)} \frac{b(\mathbf{v}, p)}{\|\mathbf{v}\|_{V_0(\Omega)}} = \frac{1}{\beta} \sup_{\mathbf{v} \in V_0(\Omega)} \frac{F(\mathbf{v}) - a(\mathbf{u}, \mathbf{v})}{\|\mathbf{v}\|_{V_0(\Omega)}} \leq \\ &\leq \frac{1}{\beta} \left(1 + \frac{\gamma}{\alpha}\right) \left(\|\mathbf{f}\|_{V'(\Omega)} + C_T \|\mathbf{h}\|_{H^{-1/2}(\Gamma_{out})}\right). \end{aligned}$$

Infine (si veda ad esempio [Tem79]) è possibile mostrare il seguente risultato di regolarità: se  $\Omega$  è di classe  $C^{1,1}$  a tratti e si considerano condizioni di Dirichlet anche su  $\Gamma_{out}$ ,  $\mathbf{f} \in [L^2(\Omega)]^3$  e  $\mathbf{u}_D \in [H^{3/2}(\partial\Omega)]^3$ , allora la soluzione  $(\mathbf{u}(\Omega), p(\Omega))$  appartiene ad  $\{[H^2(\Omega)]^3 \cap V_D(\Omega)\} \times H^1(\Omega)$ . Questo risultato sarà utile per giustificare la validità delle condizioni di ottimalità ricavate nella Sezione 3.2.5.

### 3.2.2 Un risultato di esistenza per il problema di ottimizzazione di forma

In questa sezione si dimostra un risultato di esistenza per il problema di ottimizzazione di forma (3.1), mediante il Teorema 2.4 di esistenza. Si considerino i seguenti insiemi di forme ammissibili:

$$\mathcal{O}_{ad}^{(a)} = \mathcal{O}_{ad}^{(b)} = \{\Omega \in \mathcal{O}_\varepsilon : |\Omega| = \bar{V}, \Gamma_{in} \cup \Gamma_w \cup \Gamma_{out} \text{ fissato}\} \quad (3.3)$$

per le geometrie di Figura 3.1(a)-(b), e

$$\mathcal{O}_{ad}^{(c)} = \{\Omega \in \mathcal{O}_\varepsilon : \Gamma_{in} \cup \Gamma_{out} \text{ fissato}\} \quad (3.4)$$

per la geometria di Figura 3.1(c). In entrambi i casi, quindi, l'unica frontiera soggetta ad ottimizzazione è  $\Gamma_c$ . Il compatto  $B$  della definizione (2.9) è indicato in blu in ciascun caso della Figura 3.1. Uno dei principali risultati di questo capitolo è costituito dal seguente Teorema:

**Teorema 3.2.** *Sia  $\mathcal{O}_{ad}$  dato da (3.3) o da (3.4), e sia  $J(\Omega)$  dato da  $(J_{steady})$ , in cui  $(\mathbf{u}(\Omega), p(\Omega))$  è la soluzione del problema di Stokes  $(S_{steady})$ . Allora esiste almeno un dominio  $\hat{\Omega} \in \mathcal{O}_{ad}$  che risolve il problema di ottimizzazione di forma*

$$\hat{\Omega} = \arg \min_{\Omega \in \mathcal{O}_{ad}} J(\Omega).$$

**DIMOSTRAZIONE.** Occorre verificare le ipotesi del Teorema 2.4 di esistenza, ed in particolare la *compattezza di  $\mathcal{G}$*  (che sarà conseguenza della *compattezza di  $\mathcal{O}_{ad}$  in  $\mathcal{O}_\varepsilon$*  e della *continuità di  $(\mathbf{u}(\Omega), p(\Omega))$  rispetto al dominio*) e la *semicontinuità inferiore di  $J$* .

**Compattezza di  $\mathcal{O}_{ad}$  in  $\mathcal{O}_\varepsilon$ :** è conseguenza del Teorema 2.10. In basa a questo risultato, data una successione  $\{\Omega_n\}_{n \geq 1} \subset \mathcal{O}_{ad}$ , esiste una sottosuccessione (denominata sempre con indice  $n$ ) ed un dominio  $\Omega \in \mathcal{O}_\varepsilon$  tali che

$$\Omega_n \xrightarrow{H} \Omega.$$

Inoltre, grazie all'ipotesi (2.10),  $\Omega$  rispetta il vincolo di volume. Per affermare che  $\Omega \in \mathcal{O}_{ad}$  occorre ancora dimostrare che le proprietà geometriche che caratterizzano l'*inlet*, l'*outlet* ed eventuali pareti sono preservate al limite, cioè che  $\partial\Omega \setminus \Gamma_c(\Omega)$  sono fissate. Ciò è conseguenza della stabilità della convergenza di Hausdorff rispetto all'inclusione e della proprietà dell' $\varepsilon$ -cono: si veda [HP10] per la dimostrazione.

**Continuità di  $(\mathbf{u}(\Omega), p(\Omega))$  rispetto al dominio e semicontinuità inferiore di  $J$ :** per dimostrare la *shape continuity* si può procedere come nella dimostrazione del Teorema 2.12:

1. *Estrazione di una sottosuccessione convergente:* siano  $\mathbf{u}_n = \mathbf{u}(\Omega_n)$ ,  $p_n = p(\Omega_n)$ . Grazie alle condizioni di Dirichlet omogenee su  $\Gamma_0$  è possibile estendere a zero  $u_n$  in  $B \setminus \Omega_n$ ; tale estensione viene denominata  $\tilde{\mathbf{u}}_n$ . Sostituendo gli integrali in  $\Omega_n$  con integrali su  $B$  nella dimostrazione della disuguaglianza di energia si ottiene

$$\begin{aligned} \|\tilde{\mathbf{u}}_n\|_{V_D(B)} &\leq C_u, \\ \|\tilde{p}_n\|_{Q(B)} &\leq C_p, \end{aligned}$$

dove le costanti  $C_u$  e  $C_p$  dipendono da  $B$ , ma non da  $n$ . Di conseguenza, eventualmente passando ad una sottosuccessione (indicata sempre con indice  $n$ ), si ha che  $\tilde{\mathbf{u}}_n \xrightarrow{V_D(B)} \tilde{\mathbf{u}}$  e  $\tilde{p}_n \xrightarrow{Q(B)} \tilde{p}$  (Teorema di Banach-Alaoglu).

Ciò è sufficiente per mostrare anche la semicontinuità inferiore del funzionale costo, grazie alla semicontinuità inferiore della norma rispetto alla convergenza debole e ricordando che, grazie alla disuguaglianza di Poincaré,  $J(\Omega)$  è una norma equivalente in  $V_D(\Omega)$ .

Rimane da mostrare che  $(\mathbf{u}, p) = (\tilde{\mathbf{u}}|_\Omega, \tilde{p}|_\Omega)$  è soluzione di  $(S_{steady})$  nel dominio  $\Omega$ .

2.  $\mathbf{u}$  soddisfa le condizioni al contorno su  $\Gamma_D(\Omega)$ : la condizione al bordo su  $\Gamma_c$  si ricava come nella dimostrazione del Teorema 2.12. Inoltre  $\mathbf{u} = \mathbf{u}_D$  su  $\Gamma_D \setminus \Gamma_c$  in quanto  $\Gamma_D \setminus \Gamma_c$  è fissato e, per ogni  $n$ ,  $\mathbf{u}_n = \mathbf{u}_D$  su  $\Gamma_D \setminus \Gamma_c$ .
3.  $(\mathbf{u}, p)$  è soluzione del problema di Stokes in  $\Omega$ : è sufficiente passare al limite nella forma debole (3.2). Infatti, per ogni  $\phi \in [C_{\Gamma_D}^\infty(\Omega)]^3$ :

$$\int_{\Omega_n} \nu \nabla \mathbf{u}_n \cdot \nabla \tilde{\phi}|_{\Omega_n} = \int_B \nu \nabla \tilde{\mathbf{u}}_n \cdot \nabla \tilde{\phi} \xrightarrow{n \rightarrow \infty} \int_B \nu \nabla \tilde{\mathbf{u}} \cdot \nabla \tilde{\phi} = \int_\Omega \nu \nabla \mathbf{u} \cdot \nabla \phi,$$

grazie alla convergenza debole  $\tilde{\mathbf{u}}_n \xrightarrow{V_D(B)} \tilde{\mathbf{u}}$ ;

$$\int_{\Omega_n} p_n \operatorname{div} \tilde{\phi}|_{\Omega_n} = \int_B \tilde{p}_n \operatorname{div} \tilde{\phi} \xrightarrow{n \rightarrow \infty} \int_B \tilde{p} \operatorname{div} \tilde{\phi} = \int_\Omega p \operatorname{div} \phi,$$

grazie alla convergenza debole  $\tilde{p}_n \xrightarrow{Q(B)} \tilde{p}$ .

Si conclude quindi che  $(\mathbf{u}, p)$  soddisfa l'equazione di Stokes in  $\Omega$ . Analogamente si dimostra che  $\mathbf{u}$  soddisfa l'equazione di continuità in  $\Omega$ .

Le ipotesi del Teorema 2.4 di esistenza sono quindi tutte verificate (la limitatezza dal basso del funzionale costo è conseguenza della positività della norma di  $V(\Omega)$ ), e dunque il problema di ottimizzazione di forma (3.1) ammette almeno una forma ottima.  $\square$

### 3.2.3 Un risultato di simmetria per la forma ottima

In questa sezione viene mostrato che esiste almeno una forma ottima di (3.1) che gode di una proprietà di simmetria che verrà precisata in seguito. Infatti, sia i domini presentati in Figura 3.1, sia le condizioni al contorno del problema di Stokes  $S_{steady}$ , presentano un'elevata simmetria, ed appare dunque legittimo chiedersi se alcune proprietà di simmetria vengano conservate anche all'ottimo. Ad esempio, a meno di un'opportuna traslazione dell'origine, i domini di Figura 3.1(a)-(b) sono simmetrici rispetto ai piani  $x_1 = 0$ ,  $x_2 = 0$  ed  $x_3 = 0$ , ed il dominio Figura 3.1(c) è simmetrico rispetto al piano  $x_1 = 0$ . Il risultato di simmetria è conseguenza del seguente Lemma:

**Lemma 3.3 ([Sal08]).** *Si consideri uno spazio di Hilbert  $X$ , una forma bilineare  $A : X \times X \rightarrow \mathbb{R}$  simmetrica, ed un funzionale  $L \in X'$ . Allora  $u \in X$  è soluzione di*

$$A(u, v) = L(v), \quad \forall v \in X$$

se e solo se

$$E(u) = \min_{v \in X} E(v), \quad E(v) = \frac{1}{2}A(v, v) - L(v).$$

Il Lemma precedente è applicabile al problema di Stokes per lo spazio di Hilbert  $V_{\text{div}}(\Omega)$  e la forma bilineare simmetrica  $a(\mathbf{u}, \mathbf{v})$ , e permette di mostrare il seguente risultato di simmetria:

**Teorema 3.4 ([HP10]).** *Si consideri il problema ( $S_{steady}$ ) per il dominio  $\Omega = \Omega^{(b)}$  oppure  $\Omega = \Omega^{(c)}$ , dove  $\mathbf{u}_{in}$  è un profilo che gode di simmetria radiale ed  $\mathbf{f} = \mathbf{h} = \mathbf{0}$ . Allora nelle ipotesi del Teorema 3.2, valgono le seguenti conclusioni:*

- (b) *esiste almeno una forma ottima  $\widehat{\Omega}^{(b)} \in \mathcal{O}_{ad}^{(b)}$  con simmetria rispetto ad un piano contenente l'asse  $x_3$ ;*
- (c) *esiste almeno una forma ottima  $\widehat{\Omega}^{(c)} \in \mathcal{O}_{ad}^{(c)}$  con simmetria rispetto al piano  $x_1 = 0$ .*

**DIMOSTRAZIONE.** L'esistenza è chiaramente conseguenza del Teorema 3.2; rimane da mostrare la simmetria. Sia  $\widehat{\Omega}$  una forma ottima di (3.1); da essa è possibile ricavare una mappa  $\sigma(\mathbf{x})$  e due sotto-domini  $\widehat{\Omega}_1, \widehat{\Omega}_2$ , come segue:

- (b) *tra tutti i piani contenenti l'asse  $x_3$ , ne esiste almeno uno che divide  $\widehat{\Omega}$  in due sottodomini  $\widehat{\Omega}_1$  e  $\widehat{\Omega}_2$  dello stesso volume, pari a  $\frac{V}{2}$ . Sia  $\sigma(\mathbf{x})$  la trasformazione che opera la simmetria rispetto a tale piano;*
- (c) *siano  $\widehat{\Omega}_1 = \widehat{\Omega} \cap \{\mathbf{x} \in \mathbb{R}^3 : x_1 \geq 0\}$ ,  $\widehat{\Omega}_2 = \widehat{\Omega} \cap \{\mathbf{x} \in \mathbb{R}^3 : x_1 \leq 0\}$ , e si ponga  $\sigma(\mathbf{x})$  pari alla trasformazione che opera la simmetria rispetto al piano  $x_1 = 0$ .*

Indicando con  $(\widehat{\mathbf{u}}, \widehat{p})$  la soluzione del problema di Stokes in  $\widehat{\Omega}$ , si definisce  $J_i, i = 1, 2$  come

$$J_i = \frac{\nu}{2} \int_{\widehat{\Omega}_i} |\nabla \widehat{\mathbf{u}}|^2;$$

vale dunque  $J(\widehat{\Omega}) = J_1 + J_2$ . Inoltre, senza perdita di generalità, a meno di uno scambio di indici, si può supporre  $J_1 \leq J_2$ .

Posto  $\widetilde{\Omega} = \widehat{\Omega}_1 \cup \sigma(\widehat{\Omega}_1)$ , si definiscono le funzioni  $(\widetilde{\mathbf{u}}, \widetilde{p})$

$$\widetilde{\mathbf{u}}(\mathbf{x}) = \begin{cases} \widehat{\mathbf{u}}(\mathbf{x}), & \text{se } \mathbf{x} \in \widehat{\Omega}_1, \\ \widehat{\mathbf{u}}(\sigma(\mathbf{x})), & \text{se } \mathbf{x} \in \sigma(\widehat{\Omega}_1), \end{cases}$$

e

$$\widetilde{p}(\mathbf{x}) = \begin{cases} \widehat{p}(\mathbf{x}), & \text{se } \mathbf{x} \in \widehat{\Omega}_1, \\ \widehat{p}(\sigma(\mathbf{x})), & \text{se } \mathbf{x} \in \sigma(\widehat{\Omega}_1). \end{cases}$$

È chiaro che  $\tilde{\mathbf{u}} \in V_D(\tilde{\Omega})$ ,  $\tilde{p} \in Q(\tilde{\Omega})$  e  $\tilde{\mathbf{u}}$  è a divergenza nulla. Inoltre

$$\frac{\nu}{2} \int_{\tilde{\Omega}} |\nabla \tilde{\mathbf{u}}|^2 = 2J_1 \leq J_1 + J_2 = J(\hat{\Omega}). \quad (3.5)$$

Grazie al Lemma 3.3, la velocità  $\mathbf{u}$ , soluzione del problema di Stokes in un generico dominio  $\Omega$ , è definita come l'unico punto di minimo del funzionale energia

$$E(\Omega)(\mathbf{v}) = \frac{\nu}{2} \int_{\Omega} |\nabla \mathbf{v}|^2$$

sullo spazio  $V_{D,\text{div}}(\Omega) = V_D(\Omega) \cap V_{\text{div}}(\Omega)$ .

Di conseguenza, dalla (3.5) si ottiene

$$J(\tilde{\Omega}) = \min_{\mathbf{v} \in V_{D,\text{div}}(\tilde{\Omega})} E(\tilde{\Omega})(\mathbf{v}) \leq E(\tilde{\Omega})(\tilde{\mathbf{u}}) = \frac{\nu}{2} \int_{\tilde{\Omega}} |\nabla \tilde{\mathbf{u}}|^2 \leq J(\hat{\Omega}),$$

cioè anche  $\tilde{\Omega}$  è ottimo. □

La geometria di Figura 3.1(b) è stata introdotta solamente per enunciare questo risultato e non sarà usata nelle simulazioni numeriche riportate nella Sezione 5.2, per motivi che saranno chiari in seguito. Purtroppo il precedente risultato di simmetria non vale per la geometria di Figura 3.1(a): infatti, provando a ripetere il punto (b) della dimostrazione, le uniche mappe  $\sigma(\mathbf{x})$  per cui la frontiera  $\Gamma_w(\tilde{\Omega}) = \Gamma_w(\hat{\Omega}_1 \cup \sigma(\hat{\Omega}_1))$  è fissata sono quelle il cui piano di simmetria è  $x_1 = 0$ ,  $x_2 = 0$ ,  $x_1 + x_2 = 0$  oppure  $x_1 - x_2 = 0$  che, in generale, non garantiscono la validità del vincolo di volume. Tuttavia appare ragionevole supporre che, se le pareti sono poste a distanza tale da non influenzare la fisica del problema, il contorno  $\Gamma_c$  della forma ottima non dipenda sensibilmente dalla geometria di  $\Gamma_w$ , e che quindi valga lo stesso risultato di simmetria anche per la geometria di Figura 3.1(a), eventualmente limitatamente ad un intorno della frontiera del corpo immerso. Per la geometria di Figura 3.1(c) è invece possibile fissare  $\sigma(\mathbf{x})$  come la trasformazione che opera la simmetria rispetto al piano  $x_1 = 0$  se su  $\mathcal{O}_{ad}^{(c)}$  non viene imposto alcun vincolo di volume.

### 3.2.4 Calcolo della *shape derivative*

Come già visto nel Capitolo 2, per scrivere le condizioni di ottimalità<sup>2</sup> è necessario calcolare la *shape derivative* del problema di stato. Nel caso in esame, essa risulta soluzione di un'ulteriore problema di Stokes, come mostrato nel seguente Teorema:

**Teorema 3.5.** *Nelle ipotesi del Teorema 3.2, sia  $(\mathbf{u}(\Omega), p(\Omega))$  soluzione di  $(S_{steady})$  in  $\Omega \in \mathcal{O}_{ad}$ , e si consideri la perturbazione dell'identità  $\mathbf{T} = \mathbf{I} + \boldsymbol{\theta}$ . Affinchè il dominio deformato  $\mathbf{T}(\Omega)$  appartenga ad  $\mathcal{O}_{ad}$ , è necessario almeno ipotizzare che*

$$\boldsymbol{\theta} = \mathbf{0}, \quad \text{su } \partial\Omega \setminus \Gamma_c.$$

Allora la shape derivative  $(\mathbf{u}'(\Omega), p'(\Omega))$ , in direzione  $\boldsymbol{\theta}$ , è soluzione di

$$\begin{cases} -\nu \Delta \mathbf{u}'(\Omega) + \nabla p'(\Omega) = \mathbf{0}, & \text{in } \Omega, \\ \text{div } \mathbf{u}'(\Omega) = 0, & \text{in } \Omega, \\ \mathbf{u}'(\Omega) = \mathbf{0}, & \text{su } \Gamma_D \setminus \Gamma_c, \\ \mathbf{u}'(\Omega) = -\frac{\partial}{\partial \mathbf{n}} \mathbf{u}(\Omega) \boldsymbol{\theta} \cdot \mathbf{n}, & \text{su } \Gamma_c, \\ \nu \frac{\partial \mathbf{u}'(\Omega)}{\partial \mathbf{n}} - p'(\Omega) \mathbf{n} = \mathbf{0}, & \text{su } \Gamma_{out}. \end{cases} \quad (SD_{steady})$$

<sup>2</sup>... senza fare uso del metodo dei moltiplicatori di Lagrange.

DIMOSTRAZIONE. Si può procedere analogamente alla Sezione 2.3.3, evidenziando per comodità la derivazione delle condizioni di Dirichlet, dell'equazione di Stokes e dell'equazione di continuità.

**Condizioni di Dirichlet:** si ponga  $\mathbf{v} = \boldsymbol{\varphi} \in \mathcal{D}(\mathbb{R}^3; \mathbb{R}^3)$ : dalle condizioni di Dirichlet su  $\Gamma_D$  si ottiene

$$\int_{\Gamma_D} (\mathbf{u}(\Omega) - \mathbf{u}_D) \cdot \boldsymbol{\varphi} = 0.$$

Derivando questo integrale di bordo, grazie al Teorema 2.19, si ottiene

$$\int_{\Gamma_D} \mathbf{u}'(\Omega) \cdot \boldsymbol{\varphi} + \int_{\Gamma_D} \left[ \frac{\partial}{\partial \mathbf{n}} \{(\mathbf{u}(\Omega) - \mathbf{u}_D) \cdot \boldsymbol{\varphi}\} + H(\mathbf{u}(\Omega) - \mathbf{u}_D) \cdot \boldsymbol{\varphi} \right] \boldsymbol{\theta} \cdot \mathbf{n} = 0,$$

da cui, ricordando che  $\mathbf{u}(\Omega) - \mathbf{u}_D = \mathbf{0}$  su  $\Gamma_D$ ,

$$\int_{\Gamma_D} \mathbf{u}'(\Omega) \cdot \boldsymbol{\varphi} + \int_{\Gamma_D} \frac{\partial}{\partial \mathbf{n}} (\mathbf{u}(\Omega) - \mathbf{u}_D) \cdot \boldsymbol{\varphi} \boldsymbol{\theta} \cdot \mathbf{n} = 0.$$

Per l'arbitrarietà di  $\boldsymbol{\varphi}$  si ottiene

$$\mathbf{u}'(\Omega) = -\frac{\partial}{\partial \mathbf{n}} (\mathbf{u}(\Omega) - \mathbf{u}_D) \boldsymbol{\theta} \cdot \mathbf{n}, \quad \text{su } \Gamma_D.$$

Infine, dato che  $\boldsymbol{\theta} = \mathbf{0}$  su  $\Gamma_D \setminus \Gamma_c$ , si ottiene

$$\begin{aligned} \mathbf{u}'(\Omega) &= \mathbf{0}, & \text{su } \Gamma_D \setminus \Gamma_c, \\ \mathbf{u}'(\Omega) &= -\frac{\partial}{\partial \mathbf{n}} \mathbf{u}(\Omega) \boldsymbol{\theta} \cdot \mathbf{n}, & \text{su } \Gamma_c. \end{aligned}$$

**Equazione di Stokes:** ponendo  $\mathbf{v} = \boldsymbol{\phi} \in C^\infty(\Omega; \mathbb{R}^3)$  nella forma debole dell'equazione di Stokes ed utilizzando il Teorema 2.19, si ottiene che la *shape derivative*  $(\mathbf{u}'(\Omega), p'(\Omega))$  soddisfa

$$\begin{aligned} \int_{\Omega} \nu \nabla \mathbf{u}'(\Omega) : \nabla \boldsymbol{\phi} + \int_{\partial \Omega} \nu \nabla \mathbf{u}(\Omega) : \nabla \boldsymbol{\phi} \boldsymbol{\theta} \cdot \mathbf{n} - \int_{\Omega} p'(\Omega) \operatorname{div} \boldsymbol{\phi} - \int_{\partial \Omega} p(\Omega) \operatorname{div} \boldsymbol{\phi} \boldsymbol{\theta} \cdot \mathbf{n} = \\ = \int_{\partial \Omega} \mathbf{f} \cdot \boldsymbol{\phi} \boldsymbol{\theta} \cdot \mathbf{n} + \int_{\partial \Omega} \left( \frac{\partial(\mathbf{h} \cdot \boldsymbol{\phi})}{\partial \mathbf{n}} + H\mathbf{h} \cdot \boldsymbol{\phi} \right) \boldsymbol{\theta} \cdot \mathbf{n}, \end{aligned}$$

da cui

$$\begin{aligned} - \int_{\Omega} \nu \Delta \mathbf{u}'(\Omega) \cdot \boldsymbol{\phi} - \int_{\partial \Omega} \nu \frac{\partial \mathbf{u}'(\Omega)}{\partial \mathbf{n}} \cdot \boldsymbol{\phi} + \int_{\partial \Omega} \nu \nabla \mathbf{u}(\Omega) : \nabla \boldsymbol{\phi} \boldsymbol{\theta} \cdot \mathbf{n} + \\ + \int_{\Omega} \nabla p'(\Omega) \cdot \boldsymbol{\phi} - \int_{\Omega} p'(\Omega) \operatorname{div} \boldsymbol{\phi} - \int_{\partial \Omega} p(\Omega) \operatorname{div} \boldsymbol{\phi} \boldsymbol{\theta} \cdot \mathbf{n} = \\ = \int_{\partial \Omega} \mathbf{f} \cdot \boldsymbol{\phi} \boldsymbol{\theta} \cdot \mathbf{n} + \int_{\partial \Omega} \left( \frac{\partial(\mathbf{h} \cdot \boldsymbol{\phi})}{\partial \mathbf{n}} + H\mathbf{h} \cdot \boldsymbol{\phi} \right) \boldsymbol{\theta} \cdot \mathbf{n}. \end{aligned}$$

Dunque:

◦ se  $\boldsymbol{\phi} \in \mathcal{D}(\Omega; \mathbb{R}^3)$  si ottiene

$$-\nu \Delta \mathbf{u}'(\Omega) + \nabla p'(\Omega) = \mathbf{0}, \quad \text{in } \Omega;$$

◦ se  $\boldsymbol{\phi} \in \mathcal{D}(\Gamma_{out}; \mathbb{R}^3)$  si ottiene, ricordando che  $\boldsymbol{\theta} = \mathbf{0}$  su  $\Gamma_{out}$ ,

$$\nu \frac{\partial \mathbf{u}'}{\partial \mathbf{n}} - p' \mathbf{n} = \mathbf{0}, \quad \text{su } \Gamma_{out}.$$



**Equazione di continuità:** ponendo  $q = \psi \in C^\infty(\Omega)$  nella forma debole dell'equazione di continuità, e derivando rispetto alla forma, si ottiene

$$\int_{\Omega} \operatorname{div} \mathbf{u}'(\Omega) \psi + \int_{\partial\Omega} \operatorname{div} \mathbf{u}(\Omega) \psi \boldsymbol{\theta} \cdot \mathbf{n} = 0;$$

dunque:

- se  $\psi \in \mathcal{D}(\Omega)$  si ottiene

$$\operatorname{div} \mathbf{u}'(\Omega) = 0, \quad \text{in } \Omega;$$

- per  $\psi \in C^\infty(\Omega)$  si ottiene

$$\int_{\Gamma_c} \operatorname{div} \mathbf{u}(\Omega) \psi \boldsymbol{\theta} \cdot \mathbf{n} = 0, \quad \forall \psi \in C^\infty(\Omega), \quad (3.6)$$

relazione che verrà utilizzata nella semplificazione delle condizioni di ottimalità.  $\square$

### 3.2.5 Formulazione del problema aggiunto

Procedendo come nella Sezione 2.4.1, in questa sezione si mostra come utilizzare il calcolo della *shape derivative* per introdurre un opportuno problema aggiunto che permetta di esprimere in modo più conveniente la derivata di forma del funzionale costo.

**Teorema 3.6.** *Nelle ipotesi del Teorema 3.5, sia  $(\mathbf{u}(\Omega), p(\Omega))$  soluzione di  $(S_{steady})$  in  $\Omega \in \mathcal{O}_{ad}$ , e si consideri la perturbazione dell'identità  $\mathbf{T} = \mathbf{I} + \boldsymbol{\theta}$ . Allora lo shape gradient di  $(J_{steady})$ , in direzione  $\boldsymbol{\theta}$ , è dato da*

$$\nabla J(\Omega) = \left[ \nu \nabla \mathbf{z} : \nabla \mathbf{u}(\Omega) - \frac{\nu}{2} |\nabla \mathbf{u}(\Omega)|^2 \right] \mathbf{n},$$

per  $(\mathbf{z}, q)$  soluzioni del seguente problema aggiunto:

$$\begin{cases} -\nu \Delta \mathbf{z} + \nabla q = -\nu \Delta \mathbf{u}(\Omega), & \text{in } \Omega, \\ \operatorname{div} \mathbf{z} = 0, & \text{in } \Omega, \\ \mathbf{z} = \mathbf{0}, & \text{su } \Gamma_D \setminus \Gamma_c, \\ \mathbf{z} = \mathbf{0}, & \text{su } \Gamma_c, \\ \nu \frac{\partial \mathbf{z}}{\partial \mathbf{n}} - q \mathbf{n} = \nu \frac{\partial \mathbf{u}(\Omega)}{\partial \mathbf{n}}, & \text{su } \Gamma_{out}. \end{cases} \quad (A_{steady})$$

DIMOSTRAZIONE. Utilizzando il Teorema 2.19, la derivata del funzionale costo è data da

$$\begin{aligned} dJ(\Omega; \boldsymbol{\theta}) &= \nu \int_{\Omega} \nabla \mathbf{u}'(\Omega) \cdot \nabla \mathbf{u}(\Omega) + \frac{\nu}{2} \int_{\Gamma_c} |\nabla \mathbf{u}(\Omega)|^2 \boldsymbol{\theta} \cdot \mathbf{n} = \\ &= -\nu \int_{\Omega} \Delta \mathbf{u}(\Omega) \cdot \mathbf{u}'(\Omega) + \nu \int_{\Gamma_{out}} \frac{\partial \mathbf{u}(\Omega)}{\partial \mathbf{n}} \cdot \mathbf{u}'(\Omega) - \frac{\nu}{2} \int_{\Gamma_c} |\nabla \mathbf{u}(\Omega)|^2 \boldsymbol{\theta} \cdot \mathbf{n}. \end{aligned}$$

Moltiplicando per  $\mathbf{z}$  l'equazione di Stokes del sistema  $(SD_{steady})$  si ottiene

$$\mathbf{0} = \nu \int_{\Omega} \nabla \mathbf{u}'(\Omega) : \nabla \mathbf{z} - \int_{\partial\Omega} \nu \frac{\partial \mathbf{u}'(\Omega)}{\partial \mathbf{n}} \cdot \mathbf{z} - \int_{\Omega} p'(\Omega) \operatorname{div} \mathbf{z} + \int_{\partial\Omega} p'(\Omega) \mathbf{n} \cdot \mathbf{z};$$

in particolare gli ultimi tre termini sono nulli: il primo ed il terzo termine si annullano grazie alle condizioni al contorno di Dirichlet omogenee del problema aggiunto ed alle condizioni di Neumann

omogenee della *shape derivative*, il secondo è nullo grazie all'equazione di continuità per la variabile aggiunta  $\mathbf{z}$ .

Moltiplicando per  $\mathbf{u}'(\Omega)$  l'equazione di Stokes di  $(A_{steady})$  si ottiene

$$\begin{aligned} & -\nu \int_{\Omega} \Delta \mathbf{u}'(\Omega) \cdot \mathbf{u}(\Omega) = \\ & \nu \int_{\Omega} \nabla \mathbf{z} : \nabla \mathbf{u}'(\Omega) - \nu \int_{\partial\Omega} \frac{\partial \mathbf{z}}{\partial \mathbf{n}} \cdot \mathbf{u}'(\Omega) - \int_{\Omega} q \operatorname{div} \mathbf{u}'(\Omega) + \int_{\partial\Omega} q \mathbf{n} \cdot \mathbf{u}'(\Omega) = \\ & \nu \int_{\Omega} \nabla \mathbf{z} : \nabla \mathbf{u}'(\Omega) - \nu \int_{\Gamma_{out}} \frac{\partial \mathbf{u}(\Omega)}{\partial \mathbf{n}} \cdot \mathbf{u}'(\Omega) + \int_{\Gamma_c} \left[ \nu \frac{\partial \mathbf{z}}{\partial \mathbf{n}} - q \mathbf{n} \right] \cdot \frac{\partial}{\partial \mathbf{n}} \mathbf{u}(\Omega) \boldsymbol{\theta} \cdot \mathbf{n}, \end{aligned}$$

dove l'ultima uguaglianza è stata ottenuta sfruttando le condizioni al contorno del problema aggiunto e del problema della *shape derivative*, e l'equazione di continuità per  $\mathbf{u}'(\Omega)$ . Eguagliando le due espressioni così ricavate si ottiene

$$-\nu \int_{\Omega} \Delta \mathbf{u}'(\Omega) \cdot \mathbf{u}(\Omega) + \nu \int_{\Gamma_{out}} \frac{\partial \mathbf{u}(\Omega)}{\partial \mathbf{n}} \cdot \mathbf{u}'(\Omega) = \int_{\Gamma_c} \left[ \nu \frac{\partial \mathbf{z}}{\partial \mathbf{n}} - q \mathbf{n} \right] \cdot \frac{\partial}{\partial \mathbf{n}} \mathbf{u}(\Omega) \boldsymbol{\theta} \cdot \mathbf{n},$$

da cui, sostituendo nella derivata del funzionale costo,

$$\begin{aligned} dJ(\Omega; \boldsymbol{\theta}) &= -\nu \int_{\Omega} \Delta \mathbf{u}(\Omega) \cdot \mathbf{u}'(\Omega) + \nu \int_{\Gamma_{out}} \frac{\partial \mathbf{u}(\Omega)}{\partial \mathbf{n}} \cdot \mathbf{u}'(\Omega) + \frac{\nu}{2} \int_{\Gamma_c} |\nabla \mathbf{u}(\Omega)|^2 \boldsymbol{\theta} \cdot \mathbf{n} = \\ &= \int_{\Gamma_c} \left[ \nu \frac{\partial}{\partial \mathbf{n}} \mathbf{z} \cdot \frac{\partial}{\partial \mathbf{n}} \mathbf{u}(\Omega) - q \mathbf{n} \cdot \frac{\partial}{\partial \mathbf{n}} \mathbf{u}(\Omega) - \frac{\nu}{2} |\nabla \mathbf{u}(\Omega)|^2 \right] \boldsymbol{\theta} \cdot \mathbf{n}. \end{aligned}$$

Inoltre, sfruttando nella prima uguaglianza la (3.6) e, nella terza, la formula di integrazione per parti sulla superficie  $\Gamma_c$ , si ha<sup>3</sup>

$$\begin{aligned} \int_{\Gamma_c} -q \frac{\partial}{\partial \mathbf{n}} \mathbf{u}(\Omega) \cdot \mathbf{n} \boldsymbol{\theta} \cdot \mathbf{n} &= \int_{\Gamma_c} q \left( \operatorname{div} \mathbf{u}(\Omega) - \frac{\partial}{\partial \mathbf{n}} \mathbf{u}(\Omega) \cdot \mathbf{n} \right) \boldsymbol{\theta} \cdot \mathbf{n} = \\ &= \int_{\Gamma_c} \operatorname{div}_{\Gamma_c} \mathbf{u}(\Omega) q \boldsymbol{\theta} \cdot \mathbf{n} = \int_{\Gamma_c} \mathbf{u}(\Omega) \cdot \nabla_{\Gamma_c} (q \boldsymbol{\theta} \cdot \mathbf{n}) + \int_{\Gamma_c} H \boldsymbol{\theta} \cdot \mathbf{n} \mathbf{u}(\Omega) \cdot \mathbf{n} = \mathbf{0}, \end{aligned}$$

grazie alle condizioni di Dirichlet omogenee su  $\Gamma_c$  per  $\mathbf{u}(\Omega)$ .

Si ottiene quindi

$$dJ(\Omega; \boldsymbol{\theta}) = \int_{\Gamma_c} \left[ \nu \frac{\partial}{\partial \mathbf{n}} \mathbf{z} \cdot \frac{\partial}{\partial \mathbf{n}} \mathbf{u}(\Omega) - \frac{\nu}{2} |\nabla \mathbf{u}(\Omega)|^2 \right] \boldsymbol{\theta} \cdot \mathbf{n},$$

<sup>3</sup>La formula di integrazione per parti si può ricavare introducendo le seguenti nozioni di geometria differenziale:

- sia  $g \in C^1(\Omega; \mathbb{R})$ : si definisce *gradiente tangenziale* di  $g$  sulla superficie  $\Gamma \subset \partial\Omega$

$$\nabla_{\Gamma} g = \nabla g - \frac{\partial g}{\partial \mathbf{n}} \mathbf{n};$$

- sia  $\mathbf{w} \in C^1(\Omega; \mathbb{R}^n)$ : si definisce *divergenza tangenziale* di  $\mathbf{w}$  sulla superficie  $\Gamma \subset \partial\Omega$

$$\operatorname{div}_{\Gamma} \mathbf{w} = \operatorname{div} \mathbf{w} - \frac{\partial \mathbf{w}}{\partial \mathbf{n}} \cdot \mathbf{n};$$

- se  $g \in W^{1,1}(\Omega; \mathbb{R})$ ,  $\mathbf{w} \in W^{1,1}(\Omega; \mathbb{R}^n)$ , allora vale la seguente formula di integrazione per parti

$$\int_{\Gamma} \mathbf{w} \cdot \nabla_{\Gamma} g = - \int_{\Gamma} g \operatorname{div}_{\Gamma} \mathbf{w} + \int_{\Gamma} H g \mathbf{w} \cdot \mathbf{n},$$

essendo  $H = \operatorname{div}_{\Gamma} \mathbf{n}$  la curvatura media della superficie  $\Gamma$ .

da cui, ricordando la definizione di prodotto scalare “:” tra tensori<sup>4</sup> e che  $\mathbf{a} = \mathbf{0}$  su  $\Gamma_c$  implica  $\nabla \mathbf{a}|_{\Gamma_c} = \frac{\partial \mathbf{a}}{\partial \mathbf{n}} \otimes \mathbf{n}$ :

$$dJ(\Omega; \boldsymbol{\theta}) = \int_{\Gamma_c} \left[ \nu \nabla \mathbf{z} : \nabla \mathbf{u}(\Omega) - \frac{\nu}{2} |\nabla \mathbf{u}(\Omega)|^2 \right] \boldsymbol{\theta} \cdot \mathbf{n}.$$

Tale formula ha senso, per lo meno quando  $\Omega$  è di classe  $C^{1,1}$  a tratti e si considerano condizioni di Dirichlet anche su  $\Gamma_{out}$ , perché  $\mathbf{u}(\Omega), \mathbf{z} \in [H^2(\Omega)]^3$  grazie al risultato di regolarità presentato nella Sezione 3.2.1.

Lo *shape gradient* è quindi

$$\nabla J(\Omega) = \left[ \nu \nabla \mathbf{z} : \nabla \mathbf{u}(\Omega) - \frac{\nu}{2} |\nabla \mathbf{u}(\Omega)|^2 \right] \mathbf{n}. \quad \square$$

La scrittura delle condizioni di ottimalità non è ancora completa per quanto riguarda l’insieme delle forme ammissibili dato da (3.3), perché bisogna ancora imporre il vincolo di volume; al contrario, la richiesta che  $\partial\Omega \setminus \Gamma_c$  sia fissato è già garantita perché si è scelto  $\boldsymbol{\theta}$  tale che  $\boldsymbol{\theta} = \mathbf{0}$  su  $\partial\Omega \setminus \Gamma_c$ . È tuttavia facile considerare anche il vincolo di volume introducendo il seguente funzionale costo:

$$\tilde{J}(\Omega) = J(\Omega) + \xi V(\Omega),$$

essendo  $\xi > 0$  un moltiplicatore di Lagrange sul vincolo di volume  $V(\Omega) = \int_{\Omega} d\mathbf{x} = \bar{V}$ . Inoltre per una generica mappa *perturbazione dell’identità*  $\mathbf{T} = \mathbf{I} + \boldsymbol{\theta}$ , grazie agli strumenti di *shape analysis* introdotti nella Sezione 2.3.2, è possibile calcolare la derivata di  $V(\Omega)$  rispetto alla forma; essa risulta data da

$$dV(\Omega; \boldsymbol{\theta}) = \int_{\partial\Omega} \boldsymbol{\theta} \cdot \mathbf{n}.$$

È possibile dunque esprimere lo *shape gradient* del funzionale  $\tilde{J}$  come

$$\nabla \tilde{J}(\Omega) = \left[ \nu \nabla \mathbf{z} : \nabla \mathbf{u}(\Omega) - \frac{\nu}{2} |\nabla \mathbf{u}(\Omega)|^2 + \xi \right] \mathbf{n}.$$

In conclusione, il sistema delle condizioni di ottimalità è costituito da

$$\begin{cases} -\nu \Delta \mathbf{u}(\Omega) + \nabla p(\Omega) = \mathbf{f}, & \text{in } \Omega, \\ \operatorname{div} \mathbf{u}(\Omega) = 0, & \text{in } \Omega, \\ \mathbf{u}(\Omega) = \mathbf{u}_{in}, & \text{su } \Gamma_{in}, \\ \mathbf{u}(\Omega) = \mathbf{0}, & \text{su } \Gamma_0, \\ \nu \frac{\partial \mathbf{u}(\Omega)}{\partial \mathbf{n}} - p(\Omega) \mathbf{n} = \mathbf{h}, & \text{su } \Gamma_{out}, \end{cases} \quad \text{problema di stato,}$$

$$\begin{cases} -\nu \Delta \mathbf{z} + \nabla q = -\nu \Delta \mathbf{u}(\Omega), & \text{in } \Omega, \\ \operatorname{div} \mathbf{z} = 0, & \text{in } \Omega, \\ \mathbf{z} = \mathbf{0}, & \text{su } \Gamma_D \setminus \Gamma_c, \\ \mathbf{z} = \mathbf{0}, & \text{su } \Gamma_c, \\ \nu \frac{\partial \mathbf{z}}{\partial \mathbf{n}} - q \mathbf{n} = \nu \frac{\partial \mathbf{u}(\Omega)}{\partial \mathbf{n}}, & \text{su } \Gamma_{out}, \end{cases} \quad \text{problema aggiunto,}$$

$$d\tilde{J}(\Omega; \boldsymbol{\theta}) = \int_{\Gamma_c} \left[ \nu \nabla \mathbf{z} : \nabla \mathbf{u}(\Omega) - \frac{\nu}{2} |\nabla \mathbf{u}(\Omega)|^2 + \xi \right] \boldsymbol{\theta} \cdot \mathbf{n}, \quad \text{equazione di sensitività.}$$

dove, per l’insieme di forme ammissibili (3.3),  $\xi > 0$  è un moltiplicatore sul vincolo di volume, mentre, per l’insieme di forme ammissibili (3.4),  $\xi = 0$  perché non viene imposto alcun vincolo di volume.

<sup>4</sup>Dati i vettori  $\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}$ , si definisce prodotto scalare “:” nello spazio dei tensori come l’applicazione bilineare e simmetrica tale che  $(\mathbf{a} \otimes \mathbf{b}) \cdot (\mathbf{c} \otimes \mathbf{d}) = (\mathbf{a} \cdot \mathbf{c}) (\mathbf{b} \cdot \mathbf{d})$ .

### 3.2.6 Risoluzione esplicita del problema aggiunto per forzante e dato di Neumann nulli

Si consideri il problema  $(A_{steady})$ , e si ponga  $\tilde{\mathbf{z}} = \mathbf{z} - \mathbf{u}(\Omega)$ : l'equazione nella nuova variabile  $\tilde{\mathbf{z}}$  risulta data da

$$\begin{cases} -\nu\Delta\tilde{\mathbf{z}} + \nabla q = -\nu\Delta\mathbf{u}(\Omega) + \nabla p(\Omega) - \mathbf{f}, & \text{in } \Omega, \\ \operatorname{div} \tilde{\mathbf{z}} = 0, & \text{in } \Omega, \\ \tilde{\mathbf{z}} = -\mathbf{u}_{in}, & \text{su } \Gamma_{in}, \\ \tilde{\mathbf{z}} = \mathbf{0}, & \text{su } \Gamma_0, \\ \nu\frac{\partial\tilde{\mathbf{z}}}{\partial\mathbf{n}} - q\mathbf{n} = \nu\frac{\partial\mathbf{u}(\Omega)}{\partial\mathbf{n}} - p(\Omega)\mathbf{n} - \mathbf{h}, & \text{su } \Gamma_{out}; \end{cases}$$

sfruttando l'equazione di Stokes e le condizioni al contorno su  $\Gamma_{out}$  di  $(S_{steady})$  è possibile semplificare, rispettivamente, i termini noti dell'equazione aggiunta di Stokes e delle sue condizioni al contorno su  $\Gamma_{out}$ , ottenendo il seguente problema:

$$\begin{cases} -\nu\Delta\tilde{\mathbf{z}} + \nabla q = \mathbf{0}, & \text{in } \Omega, \\ \operatorname{div} \tilde{\mathbf{z}} = 0, & \text{in } \Omega, \\ \tilde{\mathbf{z}} = -\mathbf{u}_{in}, & \text{su } \Gamma_{in}, \\ \tilde{\mathbf{z}} = \mathbf{0}, & \text{su } \Gamma_0, \\ \nu\frac{\partial\tilde{\mathbf{z}}}{\partial\mathbf{n}} - q\mathbf{n} = \mathbf{0}, & \text{su } \Gamma_{out}. \end{cases} \quad (\tilde{A}_{steady})$$

Se inoltre  $\mathbf{f} = \mathbf{h} = \mathbf{0}$ , allora si verifica che  $\tilde{\mathbf{z}} = -\mathbf{u}(\Omega)$ ,  $q = -p(\Omega)$  sono soluzione di  $(\tilde{A}_{steady})$ : infatti, eseguendo questa sostituzione si ottiene il problema di stato  $(S_{steady})$ . Di conseguenza si può risolvere esplicitamente il problema aggiunto  $(A_{steady})$ : ponendo quindi  $\mathbf{z} = \mathbf{0}$ ,  $q = -p(\Omega)$ , lo *shape gradient* si semplifica in

$$\nabla J(\Omega) = \left[ -\frac{\nu}{2} |\nabla\mathbf{u}(\Omega)|^2 \right] \mathbf{n}.$$

## 3.3 Problema di Stokes non stazionario

In questa sezione viene effettuata l'analisi nel modello più complesso di un flusso descritto da un problema di stato non stazionario; in questo caso, come verrà mostrato nella Sezione 3.3.4, il problema aggiunto risulta non stazionario e *backward* in tempo.

Dati  $\mathbf{f} \in L^2(0, T; \{[H^1(B)]^3\}')$ ,  $\mathbf{u}_{in} \in L^2(0, T; [H^{1/2}(\Gamma_{in})]^3)$  e  $\mathbf{h} \in L^2(0, T; [H^{-1/2}(\Gamma_{out})]^3)$ , si consideri dunque il seguente problema di Stokes non stazionario:

$$\begin{cases} \partial_t\mathbf{u}(\Omega) - \nu\Delta\mathbf{u}(\Omega) + \nabla p(\Omega) = \mathbf{f}(t), & \text{in } \Omega \times (0, T), \\ \operatorname{div} \mathbf{u}(\Omega) = 0, & \text{in } \Omega \times (0, T), \\ \mathbf{u}(\Omega) = \mathbf{u}_{in}(t), & \text{su } \Gamma_{in} \times (0, T), \\ \mathbf{u}(\Omega) = \mathbf{0}, & \text{su } \Gamma_0 \times (0, T), \\ \nu\frac{\partial\mathbf{u}(\Omega)}{\partial\mathbf{n}} - p(\Omega)\mathbf{n} = \mathbf{h}(t), & \text{su } \Gamma_{out} \times (0, T), \\ \mathbf{u}(\Omega) = \mathbf{u}_0(\Omega), & \text{in } \Omega \times \{0\}, \end{cases} \quad (S_{unsteady})$$

ed il seguente funzionale costo

$$J(\Omega) = \frac{\nu}{2} \int_0^T \int_{\Omega} |\nabla\mathbf{u}(\Omega)|^2 \, d\mathbf{x} \, dt. \quad (J_{unsteady})$$

Per quanto riguarda il dato iniziale, verranno considerati in particolare i seguenti due casi:

1.  $\mathbf{u}_0(\Omega) = \mathbf{u}_0 \in [L^2(B)]^3$ , cioè il dato iniziale non dipende da  $\Omega$ ;

2.  $\mathbf{u}_0(\Omega)$  è soluzione del problema di Stokes stazionario ( $S_{steady}$ ) in  $\Omega$ .

Nel seguenti sezioni verrà fornita l'analisi di esistenza e la scrittura delle condizioni di ottimalità per il problema di ottimizzazione di forma

$$\widehat{\Omega} = \arg \min_{\Omega \in \mathcal{O}_{ad}} J(\Omega). \quad (3.7)$$

### 3.3.1 Formulazione debole del problema di stato

In questa sezione viene richiamata la formulazione debole del problema di Stokes non stazionario ( $S_{unsteady}$ ) e la sua buona posizione; impiegando le stesse notazioni introdotte nella Sezione 3.2.1, la formulazione debole è<sup>5</sup>:

$$\begin{aligned} & \text{trovare } (\mathbf{u}(t), p(t)), \text{ con} \\ & \mathbf{u}(t) \in L^2(0, T; V_D(\Omega)) \cap C^0([0, T]; H(\Omega)), \\ & \partial_t \mathbf{u}(t) \in L^2(0, T; V'(\Omega)), \\ & p(t) \in L^2(0, T; Q(\Omega)) : \\ & \int_{\Omega} \partial_t \mathbf{u}(t) \cdot \mathbf{v} + a(\mathbf{u}(t), \mathbf{v}) + b(\mathbf{v}, p(t)) = \langle F(t), \mathbf{v} \rangle_{V'(\Omega) \times V(\Omega)}, \quad \forall \mathbf{v} \in V_0(\Omega), \\ & b(\mathbf{u}(t), q) = \mathbf{0}, \quad \forall q \in Q(\Omega), \\ & \mathbf{u}(0) = \mathbf{u}_0. \end{aligned} \quad (3.8)$$

avendo posto  $H(\Omega) = [L^2(\Omega)]^3$ .

L'esistenza di un'unica soluzione  $\mathbf{u}(t)$  è conseguenza del metodo di Faedo-Galerkin per funzioni test appartenenti al sottospazio  $V_{\text{div}}(\Omega)$ ; inoltre è possibile ricavare una *disuguaglianza di energia* per la velocità  $\mathbf{u}(t)$  e la sua derivata  $\partial_t \mathbf{u}(t)$ . Infatti, considerando per semplicità il caso in cui  $\mathbf{u}_D(t) = \mathbf{h}(t) = \mathbf{0}$ , e scegliendo, a  $t$  fissato,  $\mathbf{v} = \mathbf{u}(t)$  nella (3.8), si ottiene

$$\frac{1}{2} \frac{d}{dt} \|\mathbf{u}(t)\|_{H(\Omega)}^2 + a(\mathbf{u}(t), \mathbf{u}(t)) = \langle F(t), \mathbf{u}(t) \rangle_{V'(\Omega) \times V(\Omega)}.$$

Dalla coercività della forma bilineare  $a(\mathbf{u}, \mathbf{u})$ , ed applicando le disuguaglianze di Cauchy-Schwarz e di Young al termine noto, si ottiene dunque

$$\frac{d}{dt} \|\mathbf{u}(t)\|_{H(\Omega)}^2 + 2\alpha \|\mathbf{u}(t)\|_{V_D(\Omega)} \leq \frac{1}{\alpha} \|\mathbf{f}(t)\|_{V'(\Omega)} + \alpha \|\mathbf{u}(t)\|_{V_D(\Omega)}.$$

Integrando tra 0 ed  $s \leq T$  si ha

$$\|\mathbf{u}(s)\|_{H(\Omega)}^2 + \alpha \int_0^s \|\mathbf{u}(t)\|_{V_D(\Omega)} dt \leq \frac{1}{\alpha} \int_0^s \|\mathbf{f}(t)\|_{V'(\Omega)} dt + \|\mathbf{u}_0\|_{H(\Omega)}^2,$$

da cui si può dedurre che  $\mathbf{u}(t) \in L^2(0, T; V_D(\Omega)) \cap L^\infty(0, T; H(\Omega))$ .

Scegliendo invece  $\mathbf{v} = \partial_t \mathbf{u}_m(t)$ , essendo  $\mathbf{u}_m(t)$  la  $m$ -esima approssimazione di Faedo-Galerkin della soluzione  $\mathbf{u}(t)$ , e passando al limite per  $m \rightarrow \infty$ , si ottiene

$$\begin{aligned} \int_{\Omega} |\partial_t \mathbf{u}(t)|^2 + \frac{d}{dt} a(\mathbf{u}(t), \mathbf{u}(t)) &= \langle F(t), \partial_t \mathbf{u}(t) \rangle_{V'(\Omega) \times V(\Omega)} \leq \\ &\leq \frac{1}{2} \|f(t)\|_{V'(\Omega)}^2 + \frac{1}{2} \int_{\Omega} |\partial_t \mathbf{u}(t)|^2, \end{aligned}$$

<sup>5</sup>Con abuso di notazione, spesso si indicherà con  $\int_{\Omega} \partial_t \mathbf{u}(t) \cdot \mathbf{v}$  la dualità  $\langle \partial_t \mathbf{u}(t), \mathbf{v} \rangle_{V'(\Omega) \times V(\Omega)}$

da cui, integrando tra 0 e  $T$  si ottiene

$$\int_0^T \int_{\Omega} |\partial_t \mathbf{u}(t)|^2 + 2 \int_0^T a(\mathbf{u}(T), \mathbf{u}(T)) \leq \int_0^T \|f(t)\|_{V'(\Omega)}^2,$$

cioè  $\partial_t \mathbf{u}(t) \in L^2(0, T; L^2(\Omega))$ , ed in particolare  $\partial_t \mathbf{u}(t) \in L^2(0, T; V'(\Omega))$ .

Applicando quindi il Lemma 3.1 al funzionale  $L$  dato da

$$L(\mathbf{v}) = \langle F(t), \mathbf{v} \rangle_{V'(\Omega) \times V(\Omega)} - \int_{\Omega} \partial_t \mathbf{u}(t) \cdot \mathbf{v} - a(\mathbf{u}(t), \mathbf{v}) \in V'(\Omega),$$

si ottiene l'esistenza di un'unica pressione  $p(t)$  tale che  $(\mathbf{u}(t), p(t))$  è soluzione di (3.8). Si può mostrare, analogamente al caso stazionario, che  $p(t) \in L^2(0, T; Q(\Omega))$ , utilizzando la condizione *inf-sup*; infatti

$$\begin{aligned} \|p(t)\|_{Q(\Omega)} &\leq \frac{1}{\beta} \sup_{\mathbf{v} \in V_0(\Omega)} \frac{b(\mathbf{v}, p(t))}{\|\mathbf{v}\|_{V_0(\Omega)}} = \\ &= \frac{1}{\beta} \sup_{\mathbf{v} \in V_0(\Omega)} \frac{\langle F(t), \mathbf{v} \rangle_{V'(\Omega) \times V(\Omega)} - \int_{\Omega} \partial_t \mathbf{u}(t) \cdot \mathbf{v} - a(\mathbf{u}(t), \mathbf{v})}{\|\mathbf{v}\|_{V_0(\Omega)}} \leq \\ &\leq \frac{1}{\beta} \left( \|f(t)\|_{V'(\Omega)} + \|\partial_t \mathbf{u}(t)\|_{V'(\Omega)} + \gamma \|\mathbf{u}(t)\|_{V_D(\Omega)} \right), \end{aligned}$$

da cui, elevando al quadrato ed integrando in tempo, si ottiene

$$\begin{aligned} \|p(t)\|_{L^2(0, T; Q(\Omega))}^2 &\leq \frac{1}{\beta^2} \left( 2 \|f(t)\|_{L^2(0, T; V'(\Omega))}^2 + 2 \|\partial_t \mathbf{u}(t)\|_{L^2(0, T; V'(\Omega))}^2 + \right. \\ &\quad \left. + 2\gamma^2 \|\mathbf{u}(t)\|_{L^2(0, T; V_D(\Omega))}^2 \right). \end{aligned}$$

Sostituendo le disuguaglianze dell'energia per  $\mathbf{u}(t)$  e  $\partial_t \mathbf{u}(t)$  si ottiene la stima desiderata.

Infine (si veda ad esempio [Tem79]) è possibile mostrare il seguente risultato di regolarità, analogo a quello ottenuto per il caso stazionario: se  $\Omega$  è di classe  $C^{1,1}$ , si considerano condizioni di Dirichlet anche su  $\Gamma_{out}$ ,  $f \in L^2(0, T; H(\Omega))$  e  $\mathbf{u}_D \in L^2(0, T; [H^{3/2}(\partial\Omega)]^3)$ , allora la soluzione  $(\partial_t \mathbf{u}(\Omega), \mathbf{u}(\Omega), p(\Omega))$  appartiene ad  $L^2(0, T; H(\Omega)) \times L^2(0, T; [H^2(\Omega)]^3 \cap V_D(\Omega)) \times L^2(0, T; H^1(\Omega))$ .

### 3.3.2 Un risultato di esistenza per il problema di ottimizzazione di forma

In questa sezione si dimostra il seguente risultato di esistenza per il problema di ottimizzazione di forma (3.7), ricorrendo al risultato generale di esistenza fornito dal Teorema 2.4.

**Teorema 3.7.** *Sia  $\mathcal{O}_{ad}$  dato da (3.3) o da (3.4), e sia  $J(\Omega)$  dato da  $(J_{unsteady})$ , in cui  $(\mathbf{u}(\Omega), p(\Omega))$  è soluzione del problema di Stokes non stazionario  $(S_{unsteady})$ . Se il dato iniziale  $\mathbf{u}_0(\cdot)$  è tale che, per ogni successione di domini  $\{\Omega_n\}_{n \geq 1} \subset \mathcal{O}_{ad}$  e per ogni dominio  $\Omega \in \mathcal{O}_{ad}$  tali che  $\Omega_n \xrightarrow{H} \Omega$ , vale*

$$\tilde{\mathbf{u}}_0(\Omega_n) \xrightarrow{H(B)} \tilde{\mathbf{u}}_0(\Omega), \quad (3.9)$$

avendo indicato con  $\tilde{\mathbf{u}}_0(\Omega)$  l'estensione a zero in  $B \setminus \Omega$ , allora esiste almeno un dominio  $\hat{\Omega} \in \mathcal{O}_{ad}$  che risolve il problema di ottimizzazione di forma

$$\hat{\Omega} = \arg \min_{\Omega \in \mathcal{O}_{ad}} J(\Omega).$$

DIMOSTRAZIONE. Occorre verificare le ipotesi del Teorema 2.4 di esistenza, ed in particolare la *compattezza di  $\mathcal{G}$*  (che sarà conseguenza della *compattezza di  $\mathcal{O}_{ad}$  in  $\mathcal{O}_\varepsilon$*  e della *continuità di  $(\mathbf{u}(\Omega), p(\Omega))$  rispetto al dominio*) e la *semicontinuità inferiore di  $J$* .

**Compattezza di  $\mathcal{O}_{ad}$  in  $\mathcal{O}_\varepsilon$ :** la dimostrazione è invariata rispetto a quella del Teorema 3.2.

**Continuità di  $(\mathbf{u}(\Omega), p(\Omega))$  rispetto al dominio e semicontinuità inferiore di  $J$ :** per dimostrare la *shape continuity* si può procedere come nelle dimostrazioni dei Teoremi 2.12 e 3.2:

1. *Estrazione di una sottosuccessione convergente:* indicando con  $\mathbf{u}_n(t) = \mathbf{u}(\Omega_n)(t)$ ,  $p_n(t) = p(\Omega_n)(t)$ , si procede analogamente alla dimostrazione del Teorema 3.2, estendendo a zero su  $B \setminus \Omega_n$  ed ottenendo, grazie alle stime di stabilità, le seguenti tre successioni debolmente convergenti

$$\begin{aligned}\tilde{\mathbf{u}}_n(t) &\xrightarrow{L^2(0,T;V_D(B))} \tilde{\mathbf{u}}(t), \\ \partial_t \tilde{\mathbf{u}}_n(t) &\xrightarrow{L^2(0,T;V'(B))} D(t), \\ \tilde{p}_n(t) &\xrightarrow{L^2(0,T;Q(B))} \tilde{p}(t).\end{aligned}$$

Inoltre  $D(t) = \partial_t \tilde{\mathbf{u}}(t)$  perché, se  $\phi \in [C_{\Gamma_D}^\infty(\Omega)]^3$  e  $\psi(t) \in \mathcal{D}(\Omega)$ , vale la seguente formula di integrazione per parti:

$$\int_0^T \int_{\Omega_n} \partial_t \mathbf{u}_n(t) \cdot \tilde{\phi}|_{\Omega_n} \psi(t) = - \int_0^T \int_{\Omega_n} \mathbf{u}_n(t) \cdot \tilde{\phi}|_{\Omega_n} \partial_t \psi(t),$$

la quale, riscritta come

$$\int_0^T \int_B \partial_t \tilde{\mathbf{u}}_n(t) \cdot \tilde{\phi}|_{\Omega_n} \psi(t) = - \int_0^T \int_B \tilde{\mathbf{u}}_n(t) \cdot \tilde{\phi}|_{\Omega_n} \partial_t \psi(t),$$

è valida anche al limite grazie alle convergenze deboli in  $L^2(0,T;V'(B))$  e  $L^2(0,T;V_D(B))$ , rispettivamente.

Ciò è sufficiente per mostrare anche la semicontinuità inferiore del funzionale costo, grazie alla semicontinuità inferiore della norma rispetto alla convergenza debole e ricordando che, grazie alla disuguaglianza di Poincaré,  $J(\Omega)$  è una norma equivalente in  $L^2(0,T;V_D(\Omega))$ .

Rimane da mostrare che  $(\partial_t \mathbf{u}(t), \mathbf{u}(t), p(t)) = (\partial_t \tilde{\mathbf{u}}(t)|_\Omega, \tilde{\mathbf{u}}(t)|_\Omega, \tilde{p}(t)|_\Omega)$  è soluzione di  $(S_{unsteady})$  nel dominio  $\Omega$ .

2.  $\mathbf{u}(t)$  soddisfa le condizioni al contorno su  $\Gamma_D(\Omega) \times (0, T)$ : la dimostrazione è invariata rispetto a quella del Teorema 3.2.
3.  $(\mathbf{u}(t), p(t))$  è soluzione del problema di Stokes non stazionario in  $\Omega \times (0, T)$ : è sufficiente passare al limite nella forma debole. Infatti, per ogni  $\phi \in [C_{\Gamma_D}^\infty(\Omega)]^3$ ,  $\psi(t) \in \mathcal{D}(0, T)$ :

$$\begin{aligned}\int_0^T \int_{\Omega_n} \partial_t \mathbf{u}_n(t) \cdot \tilde{\phi}|_{\Omega_n} \psi(t) &= \int_0^T \int_B \partial_t \tilde{\mathbf{u}}_n(t) \cdot \tilde{\phi} \psi(t) \xrightarrow{n \rightarrow \infty} \\ &\xrightarrow{n \rightarrow \infty} \int_0^T \int_B \partial_t \tilde{\mathbf{u}}(t) \cdot \tilde{\phi} \psi(t) = \int_0^T \int_\Omega \partial_t \mathbf{u}(t) \cdot \phi \psi(t),\end{aligned}$$

grazie alla convergenza debole  $\partial_t \tilde{\mathbf{u}}_n(t) \xrightarrow{L^2(0,T;V'(B))} \partial_t \tilde{\mathbf{u}}(t)$ ;

$$\begin{aligned}\int_0^T \int_{\Omega_n} \nu \nabla \mathbf{u}_n(t) \cdot \nabla \tilde{\phi}|_{\Omega_n} \psi(t) &= \int_0^T \int_B \nu \nabla \tilde{\mathbf{u}}_n(t) \cdot \nabla \tilde{\phi} \psi(t) \xrightarrow{n \rightarrow \infty} \\ &\xrightarrow{n \rightarrow \infty} \int_0^T \int_B \nu \nabla \tilde{\mathbf{u}}(t) \cdot \nabla \tilde{\phi} \psi(t) = \int_0^T \int_\Omega \nu \nabla \mathbf{u}(t) \cdot \nabla \phi \psi(t),\end{aligned}$$

grazie alla convergenza debole  $\tilde{\mathbf{u}}_n(t) \xrightarrow{L^2(0,T;V_D(B))} \tilde{\mathbf{u}}(t)$ ;

$$\begin{aligned} \int_0^T \int_{\Omega_n} p_n \operatorname{div} \tilde{\boldsymbol{\phi}}|_{\Omega_n} \psi(t) &= \int_0^T \int_B \tilde{p}_n \operatorname{div} \tilde{\boldsymbol{\phi}} \psi(t) \xrightarrow{n \rightarrow \infty} \\ &\xrightarrow{n \rightarrow \infty} \int_0^T \int_B \tilde{p} \operatorname{div} \tilde{\boldsymbol{\phi}} \psi(t) = \int_0^T \int_{\Omega_n} p \operatorname{div} \boldsymbol{\phi} \psi(t), \end{aligned}$$

grazie alla convergenza debole  $\tilde{p}_n(t) \xrightarrow{L^2(0,T;Q(B))} \tilde{p}(t)$ .

Si conclude quindi che  $(\mathbf{u}(t), p(t))$  soddisfa l'equazione di Stokes in  $\Omega \times (0, T)$ . Analogamente si dimostra che  $\mathbf{u}(t)$  soddisfa l'equazione di continuità in  $\Omega \times (0, T)$ .

4.  $\mathbf{u}(t)$  soddisfa le condizioni iniziali su  $\Omega \times \{0\}$ : per ogni  $\boldsymbol{\phi} \in [C_{\Gamma_D}^\infty(\Omega)]^3$ ,  $\psi(t) \in C^\infty(0, T)$  tale che  $\psi(T) = 0$ , si consideri nuovamente la formula di integrazione per parti:

$$\int_0^T \int_{\Omega_n} \partial_t \mathbf{u}_n(t) \cdot \tilde{\boldsymbol{\phi}}|_{\Omega_n} \psi(t) = - \int_{\Omega_n} \mathbf{u}_n(0) \cdot \tilde{\boldsymbol{\phi}}|_{\Omega_n} \psi(0) - \int_0^T \int_{\Omega_n} \mathbf{u}_n(t) \cdot \tilde{\boldsymbol{\phi}}|_{\Omega_n} \partial_t \psi(t).$$

Dato che  $\mathbf{u}_n \in C^0([0, T]; H(\Omega))$ , anche  $\tilde{\mathbf{u}}_n \in C^0([0, T]; H(B))$ : estendendo allora, come mostrato nei punti precedenti, gli integrali sul compatto fissato  $B$ , si può passare al limite per  $n \rightarrow \infty$  ed ottenere

$$\int_0^T \int_{\Omega} \partial_t \mathbf{u}(t) \cdot \boldsymbol{\phi} \psi(t) = - \int_{\Omega} \mathbf{u}(0) \cdot \boldsymbol{\phi} \psi(0) - \int_0^T \int_{\Omega} \mathbf{u}(t) \cdot \boldsymbol{\phi} \partial_t \psi(t).$$

Sostituendo tale formula di integrazione per parti in

$$\begin{aligned} \int_0^T \int_{\Omega_n} \partial_t \mathbf{u}_n(t) \cdot \tilde{\boldsymbol{\phi}}|_{\Omega_n} \psi(t) + \nu \nabla \mathbf{u}_n(t) \cdot \nabla \tilde{\boldsymbol{\phi}}|_{\Omega_n} \psi(t) - p_n \operatorname{div} \tilde{\boldsymbol{\phi}}|_{\Omega_n} \psi(t) &= \\ &= \int_0^T \int_{\Omega} F(t) \tilde{\boldsymbol{\phi}}|_{\Omega_n} \psi(t), \end{aligned} \quad (3.10)$$

e passando al limite, come già mostrato al punto 3., si ottiene

$$\begin{aligned} \int_0^T \int_{\Omega} -\mathbf{u}(t) \cdot \boldsymbol{\phi} \partial_t \psi(t) + \nu \nabla \mathbf{u}(t) \cdot \nabla \boldsymbol{\phi} \psi(t) - p \operatorname{div} \boldsymbol{\phi} \psi(t) &= \\ &= \int_{\Omega} \mathbf{u}(0) \cdot \boldsymbol{\phi} \psi(0) + \int_0^T \int_{\Omega} F(t) \boldsymbol{\phi} \psi(t). \end{aligned} \quad (3.11)$$

Tuttavia, ricordando che  $\mathbf{u}_n(0) = \mathbf{u}_0(\Omega_n)$ , vale anche

$$\int_0^T \int_{\Omega_n} \partial_t \mathbf{u}_n(t) \cdot \tilde{\boldsymbol{\phi}}|_{\Omega_n} \psi(t) = - \int_{\Omega_n} \mathbf{u}_0(\Omega_n) \cdot \tilde{\boldsymbol{\phi}}|_{\Omega_n} \psi(0) - \int_0^T \int_{\Omega_n} \mathbf{u}_n(t) \cdot \tilde{\boldsymbol{\phi}}|_{\Omega_n} \partial_t \psi(t),$$

ed è possibile passare al limite sfruttando, per il primo addendo del membro di destra, l'ipotesi (3.9). Si ottiene quindi, sostituendo in (3.10) e passando al limite,

$$\begin{aligned} \int_0^T \int_{\Omega} -\mathbf{u}(t) \cdot \boldsymbol{\phi} \partial_t \psi(t) + \nu \nabla \mathbf{u}(t) \cdot \nabla \boldsymbol{\phi} \psi(t) - p \operatorname{div} \boldsymbol{\phi} \psi(t) &= \\ &= \int_{\Omega} \mathbf{u}_0(\Omega) \cdot \boldsymbol{\phi} \psi(0) + \int_0^T \int_{\Omega} F(t) \boldsymbol{\phi} \psi(t), \end{aligned}$$

e dal confronto con (3.11) si ottiene che  $\mathbf{u}(0) = \mathbf{u}_0(\Omega)$  in  $\Omega$ .



Le ipotesi del Teorema 2.4 di esistenza sono quindi tutte verificate (la limitatezza dal basso del funzionale costo è conseguenza della positività della norma di  $L^2(0, T; V(\Omega))$ ), e dunque il problema di ottimizzazione di forma (3.7) ammette almeno una forma ottima.  $\square$

L'ipotesi (3.9) è verificata sia se il dato iniziale non dipende da  $\Omega$ , cioè  $\mathbf{u}_0(\Omega) = \mathbf{u}_0 \in H(B)$ , sia se  $\mathbf{u}_0(\Omega)$  è soluzione del problema di Stokes stazionario, come conseguenza della dimostrazione della *shape continuity* nel Teorema 3.2.

### 3.3.3 Calcolo della *shape derivative*

Come nel caso stazionario è possibile ricavare l'espressione della *shape derivative* del problema di stato ( $S_{unsteady}$ ); essa risulta soluzione di un analogo problema di Stokes non stazionario, come mostrato nel seguente risultato:

**Teorema 3.8.** *Nelle ipotesi del Teorema 3.7, sia  $(\mathbf{u}(\Omega), p(\Omega))$  soluzione di  $(S_{unsteady})$  in  $\Omega \in \mathcal{O}_{ad}$ , e si consideri la perturbazione dell'identità  $\mathbf{T} = \mathbf{I} + \boldsymbol{\theta}$ . Affinchè il dominio deformato  $\mathbf{T}(\Omega)$  appartenga ad  $\mathcal{O}_{ad}$ , è necessario almeno ipotizzare che*

$$\boldsymbol{\theta} = \mathbf{0} \quad \text{su } \partial\Omega \setminus \Gamma_c.$$

Allora la *shape derivative*  $(\mathbf{u}'(\Omega), p'(\Omega))$ , in direzione  $\boldsymbol{\theta}$ , è soluzione di

$$\begin{cases} \frac{\partial}{\partial t} \mathbf{u}'(\Omega) - \nu \Delta \mathbf{u}'(\Omega) + \nabla p'(\Omega) = \mathbf{0}, & \text{in } \Omega \times (0, T), \\ \operatorname{div} \mathbf{u}'(\Omega) = 0, & \text{in } \Omega \times (0, T), \\ \mathbf{u}'(\Omega) = \mathbf{0}, & \text{su } \{\Gamma_D \setminus \Gamma_c\} \times (0, T), \\ \mathbf{u}'(\Omega) = -\frac{\partial}{\partial \mathbf{n}} \mathbf{u}(\Omega) \boldsymbol{\theta} \cdot \mathbf{n}, & \text{su } \Gamma_c \times (0, T), \\ \nu \frac{\partial \mathbf{u}'(\Omega)}{\partial \mathbf{n}} - p'(\Omega) \mathbf{n} = \mathbf{0}, & \text{su } \Gamma_{out} \times (0, T), \\ \mathbf{u}'(\Omega) = \mathbf{u}'_0(\Omega), & \text{in } \Omega \times \{0\}, \end{cases} \quad (SD_{unsteady})$$

essendo  $\mathbf{u}'_0(\Omega)$  la *shape derivative* di  $\mathbf{u}_0(\Omega)$  in direzione  $\boldsymbol{\theta}$ .

**DIMOSTRAZIONE.** Si procede analogamente alla dimostrazione del Teorema 3.5, evidenziando per comodità la derivazione delle condizioni di Dirichlet, dell'equazione di Stokes, dell'equazione di continuità e della condizione iniziale.

**Condizioni di Dirichlet:** la dimostrazione è invariata rispetto a quella del Teorema 3.5.

**Equazione di Stokes:** è necessario calcolare, oltre a quanto già mostrato nella dimostrazione del Teorema 3.5, la derivata rispetto alla forma del termine

$$\int_{\Omega} \partial_t \mathbf{u}(\Omega) \cdot \boldsymbol{\phi},$$

per  $\boldsymbol{\phi} \in C^\infty(\Omega; \mathbb{R}^3)$ . Utilizzando il Teorema 2.19 si ottiene

$$\int_{\Omega} \frac{\partial}{\partial t} \mathbf{u}'(\Omega) \cdot \boldsymbol{\phi} + \int_{\partial\Omega} \partial_t \mathbf{u}(\Omega) \cdot \boldsymbol{\phi} \boldsymbol{\theta} \cdot \mathbf{n};$$

procedendo come nel Teorema 3.5 si ottiene dunque la prima equazione del sistema ( $SD_{unsteady}$ ).

**Equazione di continuità:** la dimostrazione è invariata rispetto a quella del Teorema 3.5.

**Condizione iniziale:** sia  $\boldsymbol{\phi} \in C^\infty(\Omega; \mathbb{R}^3)$ ; dalla condizione iniziale si ottiene

$$\int_{\Omega} (\mathbf{u}(\Omega)(t=0) - \mathbf{u}_0(\Omega)) \cdot \boldsymbol{\phi} = 0,$$

la cui derivata rispetto alla forma è

$$\int_{\Omega} (\mathbf{u}'(\Omega)(t=0) - \mathbf{u}'_0(\Omega)) \cdot \boldsymbol{\phi} + \int_{\partial\Omega} (\mathbf{u}(\Omega)(t=0) - \mathbf{u}_0(\Omega)) \cdot \boldsymbol{\phi} \boldsymbol{\theta} \cdot \mathbf{n} = 0;$$

la precedente relazione implica:

◦ se  $\boldsymbol{\phi} \in \mathcal{D}(\Omega; \mathbb{R}^3)$ ,

$$\mathbf{u}'(\Omega) = \mathbf{u}'_0(\Omega), \quad \text{in } \Omega \times \{0\};$$

◦ se  $\boldsymbol{\phi} \in C^\infty(\Omega; \mathbb{R}^3)$ ,

$$\int_{\Gamma_c} (\mathbf{u}(\Omega)(t=0) - \mathbf{u}_0(\Omega)) \cdot \boldsymbol{\phi} \boldsymbol{\theta} \cdot \mathbf{n} = 0, \quad \forall \boldsymbol{\phi} \in C^\infty(\Omega; \mathbb{R}^3), \quad (3.12)$$

relazione che verrà utilizzata nella semplificazione delle condizioni di ottimalità.  $\square$

### 3.3.4 Formulazione del problema aggiunto

Analogamente al caso stazionario mostrato nella Sezione 3.2.5, l'introduzione di un opportuno problema aggiunto permette di semplificare le condizioni di ottimalità. Come verrà mostrato nei prossimi due Teoremi, che si differenziano per le ipotesi sul dato iniziale  $\mathbf{u}_0$ , il problema aggiunto risulta un problema di Stokes non stazionario *backward*.

Nel seguente risultato si deriva l'espressione del problema aggiunto nel caso in cui la soluzione iniziale  $\mathbf{u}_0(\Omega)$  non dipenda da  $\Omega$ .

**Teorema 3.9.** *Nelle ipotesi del Teorema 3.8, sia  $(\mathbf{u}(\Omega), p(\Omega))$  soluzione di  $(S_{unsteady})$  in  $\Omega \in \mathcal{O}_{ad}$ , e si consideri la perturbazione dell'identità  $\mathbf{T} = \mathbf{I} + \boldsymbol{\theta}$ . Allora, se*

$$\mathbf{u}_0(\Omega) = \mathbf{u}_0 \in [L^2(B)]^3,$$

lo shape gradient di  $(J_{unsteady})$ , in direzione  $\boldsymbol{\theta}$ , è dato da

$$\nabla J(\Omega) = \left[ \nu \nabla \mathbf{z} : \nabla \mathbf{u}(\Omega) - \frac{\nu}{2} |\nabla \mathbf{u}(\Omega)|^2 \right] \mathbf{n},$$

per  $(\mathbf{z}, q)$  soluzioni del seguente problema aggiunto

$$\begin{cases} -\partial_t \mathbf{z} - \nu \Delta \mathbf{z} + \nabla q = -\nu \Delta \mathbf{u}(\Omega), & \text{in } \Omega \times (0, T), \\ \operatorname{div} \mathbf{z} = 0, & \text{in } \Omega \times (0, T), \\ \mathbf{z} = \mathbf{0}, & \text{su } \{\Gamma_D \setminus \Gamma_c\} \times (0, T), \\ \mathbf{z} = \mathbf{0}, & \text{su } \Gamma_c \times (0, T), \\ \nu \frac{\partial \mathbf{z}}{\partial \mathbf{n}} - q \mathbf{n} = \nu \frac{\partial \mathbf{u}(\Omega)}{\partial \mathbf{n}}, & \text{su } \Gamma_{out} \times (0, T), \\ \mathbf{z} = \mathbf{0}, & \text{in } \Omega \times \{T\}. \end{cases} \quad (A_{unsteady})$$

**DIMOSTRAZIONE.** La dimostrazione è sostanzialmente analoga a quella del Teorema 3.6, e dunque vengono riportate solo le novità introdotte dalla non stazionarietà del problema di stato.

Utilizzando il Teorema 2.19, la derivata del funzionale costo è data da

$$\begin{aligned} dJ(\Omega; \boldsymbol{\theta}) = & - \int_0^T \nu \int_{\Omega} \Delta \mathbf{u}(\Omega) \cdot \mathbf{u}'(\Omega) + \nu \int_0^T \int_{\Gamma_{out}} \frac{\partial \mathbf{u}(\Omega)}{\partial \mathbf{n}} \cdot \mathbf{u}'(\Omega) + \\ & - \frac{\nu}{2} \int_0^T \int_{\Gamma_c} |\nabla \mathbf{u}(\Omega)|^2 \boldsymbol{\theta} \cdot \mathbf{n}. \end{aligned}$$

Moltiplicando per  $\mathbf{z}$  l'equazione di Stokes della  $(SD_{unsteady})$ , ed effettuando le opportune semplificazioni, si ottiene

$$\mathbf{0} = \int_0^T \int_{\Omega} \frac{\partial}{\partial t} \mathbf{u}'(\Omega) \cdot \mathbf{z} + \nu \int_0^T \int_{\Omega} \nabla \mathbf{u}'(\Omega) : \nabla \mathbf{z}. \quad (3.13)$$

Moltiplicando per  $\mathbf{u}'(\Omega)$  l'equazione di Stokes di  $(A_{unsteady})$ , ed effettuando le opportune semplificazioni, si ha invece

$$\begin{aligned} & -\nu \int_0^T \int_{\Omega} \Delta \mathbf{u}'(\Omega) \cdot \mathbf{u}(\Omega) = \\ & = -\int_0^T \int_{\Omega} \frac{\partial}{\partial t} \mathbf{z} \cdot \mathbf{u}'(\Omega) + \nu \int_0^T \int_{\Omega} \nabla \mathbf{z} : \nabla \mathbf{u}'(\Omega) + \\ & -\nu \int_0^T \int_{\Gamma_{out}} \frac{\partial \mathbf{u}(\Omega)}{\partial \mathbf{n}} \cdot \mathbf{u}'(\Omega) + \int_0^T \int_{\Gamma_c} \left[ \nu \frac{\partial \mathbf{z}}{\partial \mathbf{n}} - q\mathbf{n} \right] \cdot \frac{\partial}{\partial \mathbf{n}} \mathbf{u}(\Omega) \boldsymbol{\theta} \cdot \mathbf{n}; \end{aligned} \quad (3.14)$$

in particolare l'integrazione per parti del primo addendo del membro di destra fornisce

$$\begin{aligned} -\int_0^T \int_{\Omega} \frac{\partial}{\partial t} \mathbf{z} \cdot \mathbf{u}'(\Omega) &= \int_0^T \int_{\Omega} \frac{\partial}{\partial t} \mathbf{u}'(\Omega) \cdot \mathbf{z} - \int_{\Omega} \mathbf{u}'(\Omega)|_{t=T} \cdot \mathbf{z}|_{t=T} + \\ &+ \int_{\Omega} \mathbf{u}'(\Omega)|_{t=0} \cdot \mathbf{z}|_{t=0}, \end{aligned} \quad (3.15)$$

in cui gli ultimi due termini si annullano in virtù della condizione *finale*  $\mathbf{z}|_{t=T} = \mathbf{0}$  e dell'ipotesi  $\mathbf{u}(\Omega)|_{t=0} = \mathbf{u}_0(\Omega)$  non dipende dalla forma  $\Omega$ .

Eguagliando le due espressioni (3.13)-(3.14) così ricavate si ottiene la seguente relazione

$$\begin{aligned} & -\nu \int_0^T \int_{\Omega} \Delta \mathbf{u}'(\Omega) \cdot \mathbf{u}(\Omega) + \nu \int_0^T \int_{\Gamma_{out}} \frac{\partial \mathbf{u}(\Omega)}{\partial \mathbf{n}} \cdot \mathbf{u}'(\Omega) = \\ & = \int_0^T \int_{\Gamma_c} \left[ \nu \frac{\partial \mathbf{z}}{\partial \mathbf{n}} - q\mathbf{n} \right] \cdot \frac{\partial}{\partial \mathbf{n}} \mathbf{u}(\Omega) \boldsymbol{\theta} \cdot \mathbf{n}, \end{aligned}$$

equivalente, a meno dell'ulteriore integrazione in tempo, a quella ottenuta nel caso stazionario. Eseguendo dunque le medesime semplificazioni si ottiene infine:

$$dJ(\Omega; \boldsymbol{\theta}) = \int_0^T \int_{\Gamma_c} \left[ \nu \nabla \mathbf{z} : \nabla \mathbf{u}(\Omega) - \frac{\nu}{2} |\nabla \mathbf{u}(\Omega)|^2 \right] \boldsymbol{\theta} \cdot \mathbf{n}.$$

Tale formula ha senso, per lo meno quando  $\Omega$  è di classe  $C^{1,1}$  a tratti e si considerano condizioni di Dirichlet anche su  $\Gamma_{out}$ , perché  $\mathbf{u}(\Omega), \mathbf{z} \in L^2(0, T; [H^2(\Omega)]^3)$  grazie al risultato di regolarità presentato nella Sezione 3.3.1. Lo *shape gradient* risulta dunque dato dalla seguente espressione:

$$\nabla J(\Omega) = \left[ \nu \nabla \mathbf{z} : \nabla \mathbf{u}(\Omega) - \frac{\nu}{2} |\nabla \mathbf{u}(\Omega)|^2 \right] \mathbf{n}. \quad \square$$

Nel caso in cui la condizione iniziale  $\mathbf{u}_0$  dipenda effettivamente da  $\Omega$ , il problema aggiunto risulta più complicato: infatti, con l'obiettivo di eliminare la dipendenza dalla *shape derivative* nell'espressione dello *shape gradient*, la presenza dell'ultimo termine non nullo in (3.15) rende necessaria l'introduzione di un'ulteriore variabile aggiunta rispetto a  $\mathbf{u}_0$ .

Nel caso più generale in cui il dato iniziale  $\mathbf{u}_0(\Omega)$  dipenda dalla forma, da un'ispezione della (3.15) non appare immediato quale problema aggiunto sia necessario formulare. Per questo motivo, nella dimostrazione del seguente risultato – valido se il dato iniziale è soluzione del problema di Stokes stazionario – ci si affiderà al metodo dei moltiplicatori di Lagrange; esso fornirà, oltre allo *shape gradient*, anche l'espressione del problema aggiunto “ausiliario”.

**Teorema 3.10.** *Nelle ipotesi del Teorema 3.8, sia  $(\mathbf{u}(\Omega), p(\Omega))$  soluzione di  $(S_{unsteady})$  in  $\Omega \in \mathcal{O}_{ad}$ , e si consideri la perturbazione dell'identità  $\mathbf{T} = \mathbf{I} + \boldsymbol{\theta}$ . Allora, se*

$$(\mathbf{u}_0(\Omega), p_0(\Omega)) \text{ è soluzione del problema di Stokes stazionario } (S_{steady}) \text{ in } \Omega,$$

la derivata di  $(J_{unsteady})$ , in direzione  $\boldsymbol{\theta}$ , è data da

$$\begin{aligned} dJ(\Omega; \boldsymbol{\theta}) &= \int_0^T \int_{\Gamma_c} \left( \nu \nabla \mathbf{z} : \nabla \mathbf{u}(\Omega) - \frac{\nu}{2} |\nabla \mathbf{u}(\Omega)|^2 \right) \boldsymbol{\theta} \cdot \mathbf{n} + \\ &+ \int_{\Gamma_c} \nu \nabla \mathbf{z}_0 : \nabla \mathbf{u}_0(\Omega) \boldsymbol{\theta} \cdot \mathbf{n}, \end{aligned}$$

per  $(\mathbf{z}, q)$  soluzioni del seguente problema aggiunto non stazionario

$$\begin{cases} -\partial_t \mathbf{z} - \nu \Delta \mathbf{z} + \nabla q = -\nu \Delta \mathbf{u}(\Omega), & \text{in } \Omega \times (0, T), \\ \operatorname{div} \mathbf{z} = 0, & \text{in } \Omega \times (0, T), \\ \mathbf{z} = \mathbf{0}, & \text{su } \{\Gamma_D \setminus \Gamma_c\} \times (0, T), \\ \mathbf{z} = \mathbf{0}, & \text{su } \Gamma_c \times (0, T), \\ \nu \frac{\partial \mathbf{z}}{\partial \mathbf{n}} - q \mathbf{n} = \nu \frac{\partial \mathbf{u}(\Omega)}{\partial \mathbf{n}}, & \text{su } \Gamma_{out} \times (0, T), \\ \mathbf{z} = \mathbf{0}, & \text{in } \Omega \times \{T\}, \end{cases} \quad (A_{s+u}^{unsteady})$$

e  $(\mathbf{z}_0, q_0)$  soluzioni del seguente problema aggiunto stazionario

$$\begin{cases} -\nu \Delta \mathbf{z}_0 + \nabla q_0 = \mathbf{z}|_{t=0}, & \text{in } \Omega, \\ \operatorname{div} \mathbf{z}_0 = 0, & \text{in } \Omega, \\ \mathbf{z}_0 = \mathbf{0}, & \text{su } \Gamma_D \setminus \Gamma_c, \\ \mathbf{z}_0 = \mathbf{0}, & \text{su } \Gamma_c, \\ \nu \frac{\partial \mathbf{z}_0}{\partial \mathbf{n}} - q_0 \mathbf{n} = \mathbf{0}, & \text{su } \Gamma_{out}. \end{cases} \quad (A_{s+u}^{steady})$$

DIMOSTRAZIONE. Si consideri il seguente funzionale Lagrangiano<sup>6</sup>:

$$\begin{aligned} \mathcal{L}(\tilde{\Omega}, \tilde{\mathbf{u}}, \tilde{p}, \tilde{\mathbf{u}}_0, \tilde{p}_0, \tilde{\mathbf{z}}, \tilde{q}, \tilde{\mathbf{z}}_0, \tilde{q}_0, \tilde{\boldsymbol{\lambda}}^D, \tilde{\boldsymbol{\lambda}}_0^D, \tilde{\boldsymbol{\lambda}}_0) &= \int_0^T \int_{\tilde{\Omega}} \frac{\nu}{2} |\nabla \tilde{\mathbf{u}}|^2 + \\ &- \int_0^T \int_{\tilde{\Omega}} (\partial_t \tilde{\mathbf{u}} - \nu \Delta \tilde{\mathbf{u}} + \nabla \tilde{p} - \mathbf{f}) \cdot \tilde{\mathbf{z}} + \int_0^T \int_{\tilde{\Omega}} \operatorname{div} \tilde{\mathbf{u}} \tilde{q} + \\ &+ \int_0^T \int_{\Gamma_D(\tilde{\Omega})} \tilde{\boldsymbol{\lambda}}^D \cdot (\tilde{\mathbf{u}} - \mathbf{u}_D) + \int_{\tilde{\Omega}} \tilde{\boldsymbol{\lambda}}_0 (\tilde{\mathbf{u}}(0) - \tilde{\mathbf{u}}_0) + \\ &- \int_{\tilde{\Omega}} (-\nu \Delta \tilde{\mathbf{u}}_0 + \nabla \tilde{p}_0 - \mathbf{f}(0)) \cdot \tilde{\mathbf{z}}_0 + \int_{\tilde{\Omega}} \operatorname{div} \tilde{\mathbf{u}}_0 \tilde{q}_0 + \\ &+ \int_{\Gamma_D(\tilde{\Omega})} \tilde{\boldsymbol{\lambda}}_0^D \cdot (\tilde{\mathbf{u}}_0 - \mathbf{u}_D(0)), \end{aligned}$$

<sup>6</sup>Per semplificare la notazione vengono adottate le seguenti convenzioni: una variabile 'generica', nel proprio spazio, è indicata con  $\tilde{\cdot}$ , ad esempio  $\tilde{\mathbf{u}}$ . La variabile corrispondente, all'ottimo, è indicata con  $\cdot$ , ad esempio  $\mathbf{u}$ . Si sottintende inoltre che, nel calcolo delle derivate di  $\mathcal{L}$ , esse siano valutate nei punti ottimi quando si ometta qualsiasi indicazione delle variabili in cui tale derivata è valutata.

dove

$$\begin{aligned}
\tilde{\Omega} &\subset \mathbb{R}^3, \\
\tilde{\mathbf{u}}, \tilde{\mathbf{z}} &\in W = L^2(0, T; H^1(\mathbb{R}^3)) \cap C^0([0, T]; L^2(\mathbb{R}^3)), & \tilde{p}, \tilde{q} &\in P = L^2(0, T; L^2(\mathbb{R}^3)), \\
\tilde{\mathbf{u}}_0, \tilde{\mathbf{z}}_0 &\in W_0 = H^1(\mathbb{R}^3), & \tilde{p}_0, \tilde{q}_0 &\in P_0 = L^2(\mathbb{R}^3), \\
\tilde{\boldsymbol{\lambda}}^D &\in W^D = L^2(0, T; H^1(\mathbb{R}^3)), & \tilde{\boldsymbol{\lambda}}_0^D &\in W_0^D = H^1(\mathbb{R}^3), \\
\tilde{\boldsymbol{\lambda}}_0 &\in W_0 = L^2(\mathbb{R}^3).
\end{aligned}$$

Procedendo come mostrato nella Sezione 2.4.2, si ricava che:

- le derivate parziali di  $\mathcal{L}$  rispetto a  $\tilde{\mathbf{z}}$  e  $\tilde{q}$  forniscono l'equazione di stato non stazionaria;
- le derivate parziali di  $\mathcal{L}$  rispetto a  $\tilde{\mathbf{z}}_0$  e  $\tilde{q}_0$  forniscono l'equazione di stato stazionaria (che definisce il dato iniziale  $\mathbf{u}_0(\Omega)$ );
- la derivata parziale di  $\mathcal{L}$  rispetto a  $\tilde{\boldsymbol{\lambda}}^D$  fornisce le condizioni al contorno di Dirichlet del problema di stato non stazionario;
- la derivata parziale di  $\mathcal{L}$  rispetto a  $\tilde{\boldsymbol{\lambda}}_0^D$  fornisce le condizioni al contorno di Dirichlet sulla velocità iniziale  $\mathbf{u}_0(\Omega)$ ;
- la derivata parziale di  $\mathcal{L}$  rispetto a  $\tilde{\boldsymbol{\lambda}}_0$  fornisce la condizione iniziale.

Per quanto riguarda la derivata rispetto alle altre variabili, si ha che:

- la derivata parziale di  $\mathcal{L}$  rispetto a  $\tilde{\mathbf{u}}$ , in direzione  $\boldsymbol{\phi} \in W$  è data da

$$\begin{aligned}
\left\langle \frac{\partial \mathcal{L}}{\partial \tilde{\mathbf{u}}}, \boldsymbol{\phi} \right\rangle &= \int_0^T \int_{\Omega} \nu \nabla \mathbf{u} : \nabla \boldsymbol{\phi} + \\
&\quad - \int_0^T \int_{\Omega} \partial_t \boldsymbol{\phi} \cdot \mathbf{z} + \int_0^T \int_{\Omega} \nu \Delta \boldsymbol{\phi} \cdot \mathbf{z} + \int_0^T \int_{\Omega} \operatorname{div} \boldsymbol{\phi} q + \\
&\quad + \int_0^T \int_{\Gamma_D(\Omega)} \boldsymbol{\lambda}^D \cdot \boldsymbol{\phi} + \int_{\Gamma_D(\Omega)} \boldsymbol{\lambda}_0 \cdot \boldsymbol{\phi}(0);
\end{aligned}$$

eseguendo alcune integrazioni per parti, si ottiene:

$$\begin{aligned}
\left\langle \frac{\partial \mathcal{L}}{\partial \tilde{\mathbf{u}}}, \boldsymbol{\phi} \right\rangle &= \int_0^T \int_{\Omega} -\nu \Delta \mathbf{u} \cdot \boldsymbol{\phi} + \int_0^T \int_{\partial \Omega} \nu \frac{\partial \mathbf{u}}{\partial \mathbf{n}} \cdot \boldsymbol{\phi} + \\
&\quad - \int_{\Omega} \mathbf{z}(T) \cdot \boldsymbol{\phi}(T) + \int_{\Omega} \mathbf{z}(0) \cdot \boldsymbol{\phi}(0) + \int_0^T \int_{\Omega} \partial_t \mathbf{z} \cdot \boldsymbol{\phi} + \\
&\quad + \int_0^T \int_{\Omega} \nu \Delta \mathbf{z} \cdot \boldsymbol{\phi} + \int_0^T \int_{\partial \Omega} \nu \mathbf{z} \cdot \frac{\partial \boldsymbol{\phi}}{\partial \mathbf{n}} - \int_0^T \int_{\partial \Omega} \nu \boldsymbol{\phi} \cdot \frac{\partial \mathbf{z}}{\partial \mathbf{n}} + \\
&\quad - \int_0^T \int_{\Omega} \boldsymbol{\phi} \cdot \nabla q + \int_0^T \int_{\partial \Omega} \boldsymbol{\phi} \cdot \mathbf{n} q \\
&\quad + \int_0^T \int_{\Gamma_D(\Omega)} \boldsymbol{\lambda}^D \cdot \boldsymbol{\phi} + \int_{\Gamma_D(\Omega)} \boldsymbol{\lambda}_0 \cdot \boldsymbol{\phi}(0).
\end{aligned}$$

Richiedendo che essa si annulli, si ricava, rispettivamente:

1. l'equazione di Stokes non stazionaria aggiunta, se  $\boldsymbol{\phi} \in \mathcal{D}(\Omega \times (0, T); \mathbb{R}^3)$ :

$$-\partial_t \mathbf{z} - \nu \Delta \mathbf{z} + \nabla q = -\nu \Delta \mathbf{u}, \quad \text{in } \Omega \times (0, T);$$

2. le condizioni al bordo per il problema non stazionario aggiunto, se  $\phi \in \mathcal{D}(\partial\Omega \times (0, T); \mathbb{R}^3)$  è tale che  $\phi = \mathbf{0}$  su  $\Gamma_D(\Omega) \times (0, T)$  e  $\frac{\partial\phi}{\partial\mathbf{n}} = \mathbf{0}$  su  $\Gamma_{out}(\Omega) \times (0, T)$ :

$$\int_0^T \int_{\Gamma_D(\Omega)} \nu \mathbf{z} \cdot \frac{\partial\phi}{\partial\mathbf{n}} + \int_0^T \int_{\Gamma_{out}(\Omega)} \phi \cdot \left( q\mathbf{n} - \nu \frac{\partial\mathbf{z}}{\partial\mathbf{n}} + \nu \frac{\partial\mathbf{u}}{\partial\mathbf{n}} \right) = 0,$$

da cui

$$\begin{aligned} \mathbf{z} &= \mathbf{0}, & \text{su } \Gamma_D(\Omega) \times (0, T), \\ \nu \frac{\partial\mathbf{z}}{\partial\mathbf{n}} - q\mathbf{n} &= \nu \frac{\partial\mathbf{u}}{\partial\mathbf{n}}, & \text{su } \Gamma_{out}(\Omega) \times (0, T); \end{aligned}$$

3. il valore del moltiplicatore di Lagrange  $\lambda^D$ , se  $\phi \in \mathcal{D}(\Gamma_D(\Omega) \times (0, T); \mathbb{R}^3)$ :

$$\int_0^T \int_{\Gamma_D(\Omega)} \phi \cdot \left( q\mathbf{n} - \nu \frac{\partial\mathbf{z}}{\partial\mathbf{n}} + \nu \frac{\partial\mathbf{u}}{\partial\mathbf{n}} + \lambda^D \right) = 0,$$

da cui

$$\lambda^D = \nu \frac{\partial\mathbf{z}}{\partial\mathbf{n}} - q\mathbf{n} - \nu \frac{\partial\mathbf{u}}{\partial\mathbf{n}}, \quad \text{su } \Gamma_D(\Omega) \times (0, T);$$

4. la condizione *finale* del problema non stazionario aggiunto, se  $\phi \in \mathcal{D}(\Omega \times \{0\}; \mathbb{R}^3)$ :

$$\int_{\Omega} \phi(T) \cdot \mathbf{z}(T) = 0,$$

da cui

$$\mathbf{z} = \mathbf{0}, \quad \text{in } \Omega \times \{T\};$$

5. il valore del moltiplicatore di Lagrange  $\lambda_0$ , se  $\phi \in \mathcal{D}(\Omega \times \{0\}; \mathbb{R}^3)$ :

$$\int_{\Omega} \phi(0) \cdot (\mathbf{z}(0) + \lambda_0) = 0,$$

da cui

$$\lambda_0 = -\mathbf{z}(0), \quad \text{in } \Omega \times \{0\};$$

o la derivata parziale di  $\mathcal{L}$  rispetto a  $\tilde{\mathbf{u}}_0$ , in direzione  $\phi_0 \in W_0$  è data da

$$\begin{aligned} \left\langle \frac{\partial\mathcal{L}}{\partial\tilde{\mathbf{u}}_0}, \phi_0 \right\rangle &= - \int_{\Omega} \lambda_0 \cdot \phi_0 + \\ &+ \int_0^T \int_{\Omega} \nu \Delta \phi_0 \cdot \mathbf{z}_0 + \int_0^T \int_{\Omega} \text{div } \phi_0 q_0 + \\ &+ \int_0^T \int_{\Gamma_D(\Omega)} \lambda_0^D \cdot \phi_0; \end{aligned}$$

eseguendo alcune integrazioni per parti e sostituendo il valore del moltiplicatore  $\lambda_0$ , si ottiene:

$$\begin{aligned} \left\langle \frac{\partial\mathcal{L}}{\partial\tilde{\mathbf{u}}_0}, \phi_0 \right\rangle &= \int_{\Omega} \mathbf{z}(0) \cdot \phi_0 + \\ &+ \int_{\Omega} \nu \Delta \mathbf{z}_0 \cdot \phi_0 + \int_{\partial\Omega} \nu \mathbf{z}_0 \cdot \frac{\partial\phi_0}{\partial\mathbf{n}} - \int_{\partial\Omega} \nu \phi_0 \cdot \frac{\partial\mathbf{z}_0}{\partial\mathbf{n}} + \\ &- \int_{\Omega} \phi_0 \cdot \nabla q_0 + \int_{\partial\Omega} \phi_0 \cdot \mathbf{n} q_0 \\ &+ \int_{\Gamma_D(\Omega)} \lambda_0^D \cdot \phi_0. \end{aligned}$$

Richiedendo che essa si annulli, si ricava, rispettivamente:

1. l'equazione di Stokes stazionaria aggiunta, se  $\phi_0 \in \mathcal{D}(\Omega; \mathbb{R}^3)$ :

$$-\nu \Delta \mathbf{z}_0 + \nabla q_0 = \mathbf{z}(0), \quad \text{in } \Omega \times (0, T);$$

2. le condizioni al bordo per il problema stazionario aggiunto, se  $\phi_0 \in \mathcal{D}(\partial\Omega; \mathbb{R}^3)$  è tale che  $\phi_0 = \mathbf{0}$  su  $\Gamma_D(\Omega)$  e  $\frac{\partial \phi_0}{\partial \mathbf{n}} = \mathbf{0}$  su  $\Gamma_{out}(\Omega)$ :

$$\int_{\Gamma_D(\Omega)} \nu \mathbf{z}_0 \cdot \frac{\partial \phi_0}{\partial \mathbf{n}} + \int_{\Gamma_{out}(\Omega)} \phi_0 \cdot \left( q_0 \mathbf{n} - \nu \frac{\partial \mathbf{z}_0}{\partial \mathbf{n}} \right) = 0,$$

da cui

$$\begin{aligned} \mathbf{z}_0 &= \mathbf{0}, & \text{su } \Gamma_D(\Omega), \\ \nu \frac{\partial \mathbf{z}_0}{\partial \mathbf{n}} - q_0 \mathbf{n} &= \mathbf{0}, & \text{su } \Gamma_{out}(\Omega); \end{aligned}$$

3. il valore del moltiplicatore di Lagrange  $\lambda^D$ , se  $\phi_0 \in \mathcal{D}(\Gamma_D(\Omega); \mathbb{R}^3)$ :

$$\int_{\Gamma_D(\Omega)} \phi_0 \cdot \left( q_0 \mathbf{n} - \nu \frac{\partial \mathbf{z}_0}{\partial \mathbf{n}} + \lambda_0^D \right) = 0,$$

da cui

$$\lambda_0^D = \nu \frac{\partial \mathbf{z}_0}{\partial \mathbf{n}} - q_0 \mathbf{n}, \quad \text{su } \Gamma_D(\Omega);$$

o la derivata parziale di  $\mathcal{L}$  rispetto a  $\tilde{p}$ , in direzione  $\psi \in P$  è

$$\left\langle \frac{\partial \mathcal{L}}{\partial \tilde{p}}, \psi \right\rangle = - \int_0^T \int_{\Omega} \nabla \psi \cdot \mathbf{z};$$

integrando per parti, si ottiene

$$\left\langle \frac{\partial \mathcal{L}}{\partial \tilde{p}}, \psi \right\rangle = \int_0^T \int_{\Omega} \operatorname{div} \mathbf{z} \psi - \int_0^T \int_{\partial\Omega} \psi \mathbf{z} \cdot \mathbf{n},$$

da cui l'equazione di continuità per la variabile aggiunta  $\mathbf{z}$ , se  $\psi \in \mathcal{D}(\Omega \times (0, T))$ :

$$\operatorname{div} \mathbf{z} = \mathbf{0}, \quad \text{in } \Omega \times (0, T);$$

o la derivata parziale di  $\mathcal{L}$  rispetto a  $\tilde{p}_0$ , in direzione  $\psi_0 \in P_0$  è

$$\left\langle \frac{\partial \mathcal{L}}{\partial \tilde{p}_0}, \psi_0 \right\rangle = - \int_0^T \int_{\Omega} \nabla \psi_0 \cdot \mathbf{z}_0;$$

integrando per parti, si ottiene

$$\left\langle \frac{\partial \mathcal{L}}{\partial \tilde{p}_0}, \psi_0 \right\rangle = \int_0^T \int_{\Omega} \operatorname{div} \mathbf{z}_0 \psi_0 - \int_0^T \int_{\partial\Omega} \psi_0 \mathbf{z}_0 \cdot \mathbf{n},$$

da cui l'equazione di continuità per la variabile aggiunta  $\mathbf{z}$ , se  $\psi_0 \in \mathcal{D}(\Omega)$ :

$$\operatorname{div} \mathbf{z}_0 = \mathbf{0}, \quad \text{in } \Omega;$$

o la derivata parziale di  $\mathcal{L}$  rispetto a  $\tilde{\Omega}$ , in direzione  $\boldsymbol{\theta}$  è

$$\begin{aligned} \left\langle \frac{\partial \mathcal{L}}{\partial \tilde{\Omega}}, \boldsymbol{\theta} \right\rangle &= dJ(\Omega; \boldsymbol{\theta}) = \int_0^T \int_{\Gamma_c(\Omega)} \frac{\nu}{2} |\nabla \mathbf{u}|^2 \boldsymbol{\theta} \cdot \mathbf{n} + \\ &\quad - \int_0^T \int_{\Gamma_c(\Omega)} (\partial_t \mathbf{u} - \nu \Delta \mathbf{u} + \nabla p - \mathbf{f}) \cdot \mathbf{z} \boldsymbol{\theta} \cdot \mathbf{n} + \int_0^T \int_{\Gamma_c(\Omega)} \operatorname{div} \mathbf{u} q \boldsymbol{\theta} \cdot \mathbf{n} + \\ &\quad - \int_{\Gamma_c(\Omega)} (-\nu \Delta \mathbf{u}_0 + \nabla p_0 - \mathbf{f}(0)) \cdot \mathbf{z}_0 \boldsymbol{\theta} \cdot \mathbf{n} + \int_{\Gamma_c(\Omega)} \operatorname{div} \mathbf{u}_0 q_0 \boldsymbol{\theta} \cdot \mathbf{n} + \\ &\quad + \int_0^T \int_{\Gamma_c(\Omega)} \left( \frac{\partial}{\partial \mathbf{n}} \{ \boldsymbol{\lambda}^D \cdot \mathbf{u} \} + H \boldsymbol{\lambda}^D \cdot \mathbf{u} \right) \boldsymbol{\theta} \cdot \mathbf{n} + \\ &\quad \int_{\Gamma_c(\Omega)} \left( \frac{\partial}{\partial \mathbf{n}} \{ \boldsymbol{\lambda}_0^D \cdot \mathbf{u}_0 \} + H \boldsymbol{\lambda}_0^D \cdot \mathbf{u}_0 \right) \boldsymbol{\theta} \cdot \mathbf{n}, \end{aligned}$$

evidenziando anche alcuni termini che si semplificano grazie alle condizioni al contorno. Sostituendo infine i valori dei moltiplicatori  $\boldsymbol{\lambda}^D$ ,  $\boldsymbol{\lambda}_0^D$  e  $\boldsymbol{\lambda}_0$  si ottiene

$$\begin{aligned} dJ(\Omega; \boldsymbol{\theta}) &= \int_0^T \int_{\Gamma_c(\Omega)} \left( -\frac{\nu}{2} |\nabla \mathbf{u}|^2 + \nu \nabla \mathbf{z} \cdot \nabla \mathbf{u} \right) \boldsymbol{\theta} \cdot \mathbf{n} + \\ &\quad \int_{\Gamma_c(\Omega)} \nu \nabla \mathbf{z}_0 \cdot \nabla \mathbf{u}_0 \boldsymbol{\theta} \cdot \mathbf{n} + \\ &\quad + \int_0^T \int_{\Gamma_c(\Omega)} \operatorname{div} \mathbf{u} q \boldsymbol{\theta} \cdot \mathbf{n} + \int_{\Gamma_c(\Omega)} \operatorname{div} \mathbf{u}_0 q_0 \boldsymbol{\theta} \cdot \mathbf{n} + \\ &\quad + \int_{\Gamma_c(\Omega)} \boldsymbol{\lambda}_0 (\mathbf{u}(0) - \mathbf{u}_0) \boldsymbol{\theta} \cdot \mathbf{n}, \end{aligned}$$

dove, procedendo come nella dimostrazione del Teorema 3.6, le prime due semplificazioni sono dovute a (3.6), mentre la terza è dovuta a (3.12).

Tale formula ha senso, per lo meno quando  $\Omega$  è di classe  $C^{1,1}$  a tratti e si considerano condizioni di Dirichlet anche su  $\Gamma_{out}$ , perché  $\mathbf{u}_0(\Omega), \mathbf{z}_0 \in [H^2(\Omega)]^3$  e  $\mathbf{u}(\Omega), \mathbf{z} \in L^2(0, T; [H^2(\Omega)]^3)$  grazie ai risultati di regolarità presentati nelle Sezioni 3.2.1 e 3.3.1.  $\square$

Analogamente al caso stazionario la scrittura delle condizioni di ottimalità per l'insieme delle forme ammissibili dato da (3.3) può essere completata introducendo un moltiplicatore di Lagrange per imporre il vincolo di volume.



## Capitolo 4

# Metodi numerici per problemi di ottimizzazione di forma

In questo capitolo vengono introdotti alcuni metodi numerici per la risoluzione di problemi di ottimizzazione di forma; uno schema iterativo generale per il calcolo di una forma ottima viene presentato nella Sezione 4.1, e descritto nel dettaglio nelle sezioni successive. In particolare, la Sezione 4.2 è dedicata ad un breve richiamo dell'approssimazione numerica del problema di Stokes, stazionario e non stazionario, mediante il metodo degli *elementi finiti*. Le restanti sezioni mostrano due possibili strategie di aggiornamento della forma: una prima opzione consiste nell'approccio di variazione locale del contorno (*local boundary variation*), presentato nella Sezione 4.3; esso dà luogo ad un metodo di tipo gradiente applicato direttamente ai nodi della griglia di calcolo situati sulla frontiera della forma da ottimizzare, la cui posizione viene di volta in volta aggiornata. Una seconda opzione, analizzata in dettaglio nella Sezione 4.4 ed applicata nelle simulazioni numeriche che verranno presentate nel Capitolo 5, è basata invece sulla mappa *Free Form Deformation*, introdotta nella Sezione 2.3.1. Dopo aver mostrato come ottenere le condizioni di ottimalità in funzione dei parametri della mappa (Sezione 4.4.1), ci si può ricondurre ad un problema di ottimizzazione che coinvolge un numero ridotto di variabili, e che può essere risolto, ad esempio, tramite il metodo del gradiente con proiezione presentato nella Sezione 4.4.2. Lo schema iterativo con parametrizzazione della forma secondo *Free Form Deformation* è stato inoltre implementato in un codice C++, sviluppato nell'ambito della libreria *LifeV*; alcuni dettagli implementativi sono mostrati nella Sezione 4.4.4 e, con maggiore dettaglio, nell'Appendice A.

### 4.1 Schema iterativo per il calcolo di una forma ottima

Questa sezione è dedicata alla presentazione di uno schema iterativo, basato su un approccio di tipo *ottimizza-e-discretizza* (si veda ad esempio [Qua08]), in cui si procede prima alla scrittura delle condizioni di ottimalità e poi alla discretizzazione del relativo sistema di EDP. Al contrario, utilizzando il paradigma *discretizza-e-ottimizza* si introduce prima la discretizzazione del problema di stato e poi si ricavano le condizioni di ottimalità sul problema di stato discreto.

Un metodo per la ricerca di una forma ottima, a partire da una configurazione iniziale  $\Omega^{(0)}$ , può essere schematizzato mediante il seguente algoritmo iterativo:

- (1) discretizzare e risolvere il problema di stato sul dominio  $\Omega^{(n)}$ , e ricavare le corrispondenti variabili di stato  $u_h^{(n)}$ ;
- (2) valutare il funzionale costo  $J_h(\Omega^{(n)}) = J(\Omega^{(n)}, u_h^{(n)})$ ;
- (3) discretizzare e risolvere il problema aggiunto sul dominio  $\Omega^{(n)}$ , e ricavare le corrispondenti variabili aggiunte  $p_h^{(n)}$ ;

- (4) calcolare lo *shape gradient*  $\nabla J_h(\Omega^{(n)}) = \nabla J(\Omega^{(n)}, u_h^{(n)}, p_h^{(n)})$ ;
- (5) estrarre dallo *shape gradient* le informazioni per l'aggiornamento della forma in un nuovo dominio  $\Omega^{(n+1)}$ ,

fino a che un criterio di arresto non è soddisfatto. In particolare, si può terminare l'algoritmo quando una opportuna norma dello *shape gradient* è minore di una tolleranza fissata, oppure quando la variazione del funzionale costo tra una iterazione e la successiva è trascurabile, oppure dopo che sono state eseguite un numero sufficiente di iterazioni.

La risoluzione numerica dei problemi (1) e (3) viene affrontata, per quanto riguarda gli esempi in ambito fluidodinamico presentati nel Capitolo 3, nella prossima sezione; le sezioni successive saranno invece dedicate all'approfondimento di due possibili scelte riguardanti l'aggiornamento della forma (5).

## 4.2 Approssimazione del problema di stato e del problema aggiunto

In questa sezione viene brevemente richiamata l'approssimazione ad *elementi finiti* dei problemi di stato ed aggiunto introdotti nel capitolo precedente; per ulteriori dettagli si rimanda, ad esempio, a [GR86].

### 4.2.1 Problema di Stokes stazionario

Si consideri il problema di Stokes stazionario ( $S_{steady}$ ) e, con le notazioni della Sezione 3.2.1, si introducano i seguenti spazi ad elementi finiti:

$$\begin{aligned} Q_h(\Omega) &\subset Q(\Omega), & V_h(\Omega) &\subset V(\Omega), \\ V_{h,D}(\Omega) &= \{\mathbf{v}_h \in V_h(\Omega) : \mathbf{v}_h|_{\Gamma_D} = \mathbf{u}_{h,D}, \mathbf{v}_h|_{\Gamma_0} = \mathbf{0}\}, & V_{h,0}(\Omega) &= V_h(\Omega) \cap V_0(\Omega), \end{aligned}$$

essendo  $\mathbf{u}_{h,D}$  una opportuna interpolazione del dato al bordo  $\mathbf{u}_D$ . L'approssimazione di Galerkin del problema ( $S_{steady}$ ) è quindi:

$$\begin{aligned} \text{trovare } \mathbf{u}_h \in V_{h,D}(\Omega), p_h \in Q_h(\Omega) : \\ a(\mathbf{u}_h, \mathbf{v}_h) + b(\mathbf{v}_h, p_h) &= F(\mathbf{v}_h), & \forall \mathbf{v}_h \in V_{h,0}(\Omega), \\ b(\mathbf{u}_h, q_h) &= \mathbf{0}, & \forall q_h \in Q_h(\Omega). \end{aligned} \quad (4.1)$$

Analogamente a quanto descritto per il caso continuo, si può mostrare che vale il seguente risultato di buona posizione:

**Teorema 4.1 ([Qua08]).** *Se la forma bilineare  $a(\mathbf{u}_h, \mathbf{v}_h)$  è continua, di costante  $\gamma$ , e coerciva su  $V_{h,0}$  di costante  $\alpha_h$ , e la forma bilineare  $b(\mathbf{v}_h, q_h)$  è continua, di costante  $\delta$ , e verifica la seguente condizione inf-sup discreta*

$$\exists \beta_h > 0 : \quad \inf_{q_h \in Q_h(\Omega)} \sup_{\mathbf{v}_h \in V_{h,0}(\Omega)} \frac{b(\mathbf{v}_h, q_h)}{\|\mathbf{v}_h\|_{V_0(\Omega)} \|q_h\|_{Q(\Omega)}} \geq \beta_h, \quad (4.2)$$

dove  $\beta_h \rightarrow \beta > 0$  se  $h \rightarrow 0$ , allora per ogni  $F \in V'(\Omega)$  esiste un'unica soluzione  $(\mathbf{u}_h, p_h)$  del problema (4.1), la quale soddisfa le seguenti stime di stabilità

$$\|\mathbf{u}_h\|_{V_D(\Omega)} \leq \frac{1}{\alpha_h} \|\mathbf{F}\|_{V'(\Omega)}, \quad \|p_h\|_{Q(\Omega)} \leq \frac{1}{\beta_h} \left(1 + \frac{\gamma}{\alpha_h}\right) \|\mathbf{F}\|_{V'(\Omega)}.$$

Inoltre, indicando con  $(\mathbf{u}, p)$  la soluzione del problema continuo (3.2), valgono le seguenti stime dell'errore

$$\begin{aligned} \|\mathbf{u} - \mathbf{u}_h\|_{V(\Omega)} &\leq \left(1 + \frac{\gamma}{\alpha_h}\right) \inf_{\mathbf{v}_h^* \in V_{h,div}(\Omega)} \|\mathbf{u} - \mathbf{v}_h^*\|_{V(\Omega)} + \frac{\delta}{\alpha_h} \inf_{q_h \in Q_h(\Omega)} \|p - q_h\|_{Q(\Omega)}, \\ \|p - p_h\|_{Q(\Omega)} &\leq \frac{\gamma}{\beta_h} \left(1 + \frac{\gamma}{\alpha_h}\right) \inf_{\mathbf{v}_h^* \in V_{h,div}(\Omega)} \|\mathbf{u} - \mathbf{v}_h^*\|_{V(\Omega)} + \\ &\quad + \left(1 + \frac{\delta}{\beta_h} + \frac{\gamma\delta}{\alpha_h\beta_h}\right) \inf_{q_h \in Q_h(\Omega)} \|p - q_h\|_{Q(\Omega)}, \end{aligned}$$

essendo  $V_{h,div}(\Omega)$  lo spazio delle funzioni a divergenza discreta nulla, cioè

$$V_{h,div}(\Omega) = \{\mathbf{v}_h \in V_h(\Omega) : b(\mathbf{v}_h, q_h) = 0, \quad \forall q_h \in Q_h(\Omega)\}.$$

La condizione *inf-sup* discreta permette inoltre di ricavare la seguente stima

$$\inf_{\mathbf{v}_h^* \in V_{h,div}(\Omega)} \|\mathbf{u} - \mathbf{v}_h^*\| \leq \left(1 + \frac{\delta}{\beta_h}\right) \inf_{\mathbf{v}_h \in V_h(\Omega)} \|\mathbf{u} - \mathbf{v}_h\|.$$

È ben noto che la validità della condizione *inf-sup* discreta (4.2) si ripercuote sulla scelta di coppie  $(V_h(\Omega), Q_h(\Omega))$  compatibili (o *inf-sup* stabili): indicando con  $\mathcal{T}_h(\Omega)$  la triangolazione di  $\Omega \subset \mathbb{R}^3$  per mezzo di elementi tetraedrici e con  $X_h^r(\Omega)$  lo spazio ad elementi finiti

$$X_h^r(\Omega) = \{v \in C^0(\Omega) : v|_K \in \mathbb{P}^r(K), \quad \forall K \in \mathcal{T}_h(\Omega)\},$$

si può dimostrare che le seguenti coppie soddisfano la condizione *inf-sup* discreta (4.2):

- elementi finiti  $\mathbb{P}_{bolla}^1 - \mathbb{P}^1$ , cioè  $(V_h(\Omega), Q_h(\Omega)) = ([X_h^1(\Omega) \oplus B_h(\Omega)]^3, X_h^1(\Omega))$ , dove  $B_h(\Omega) \subset H_0^1(\Omega)$  può indicare lo spazio delle bolle lineari a tratti o cubiche;
- elementi finiti  $\mathbb{P}^2 - \mathbb{P}^1$  (*Taylor-Hood*), cioè  $(V_h(\Omega), Q_h(\Omega)) = ([X_h^2(\Omega)]^3, X_h^1(\Omega))$ .

Indicando quindi con  $\{\phi_j \in V_h(\Omega)\}_{j=1,\dots,3 \cdot N_u}$  e  $\{\psi_k \in Q_h(\Omega)\}_{k=1,\dots,N_p}$  le funzioni di base per gli spazi  $V_h(\Omega)$  e  $Q_h(\Omega)$ , rispettivamente, la soluzione discreta  $(\mathbf{u}_h, p_h)$  può essere espressa come

$$\mathbf{u}_h(\mathbf{x}) = \sum_j u_j \phi_j(\mathbf{x}), \quad p_h(\mathbf{x}) = \sum_k p_k \psi_k(\mathbf{x});$$

scegliendo dunque, nella (4.1), le funzioni test  $\mathbf{v}_h = \phi_j$  e  $q_h = \psi_k$ , si ottiene il seguente sistema lineare:

$$\begin{bmatrix} A & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} \mathbf{U} \\ \mathbf{P} \end{bmatrix} = \begin{bmatrix} \mathbf{F} \\ \mathbf{0} \end{bmatrix}, \quad (4.3)$$

essendo  $A = [a(\phi_j, \phi_i)]_{ij}$  la matrice di *stiffness*,  $B = [b(\phi_j, \psi_k)]_{kj}$  la matrice corrispondente al termine di pressione-divergenza,  $\mathbf{F} = [F(\phi_j)]_j$  il vettore associato al termine noto ed  $\mathbf{U} \in \mathbb{R}^{3 \cdot N_u}$ ,  $\mathbf{P} \in \mathbb{R}^{N_p}$  i vettori contenenti rispettivamente i gradi di libertà  $\{u_j\}_{j=1,\dots,3 \cdot N_u}$  e  $\{p_k\}_{k=1,\dots,N_p}$ . Una volta risolto, il sistema (4.3) fornisce la soluzione approssimata del problema di Stokes stazionario ( $S_{steady}$ ).

### 4.2.2 Problema di Stokes non stazionario

Con le notazioni della Sezione 3.3.1, la formulazione di Galerkin del problema ( $S_{unsteady}$ ) può essere scritta come segue:

$$\begin{aligned} &\text{trovare } \mathbf{u}_h(t) \in V_{h,D}(\Omega), p_h(t) \in Q_h(\Omega) : \\ &\int_{\Omega} \partial_t \mathbf{u}_h(t) \cdot \mathbf{v}_h + a(\mathbf{u}_h(t), \mathbf{v}_h) + b(\mathbf{v}_h, p_h(t)) = \langle F(t), \mathbf{v}_h \rangle_{V'(\Omega) \times V(\Omega)}, \quad \forall \mathbf{v}_h \in V_{h,0}(\Omega), \\ & b(\mathbf{u}_h(t), q_h) = 0, \quad \forall q_h \in Q_h(\Omega), \\ & \mathbf{u}_h(0) = \mathbf{u}_{h,0}, \end{aligned}$$

essendo  $\mathbf{u}_{h,0}$  una opportuna interpolazione del dato iniziale  $\mathbf{u}_0$ .

Suddividendo l'intervallo  $(0, T)$  in  $N$  sottointervalli  $I^n = (t^n, t^{n+1})$ ,  $n = 0, \dots, N$ , e supponendo per semplicità  $t^{n+1} - t^n = \Delta t$  costante, si può discretizzare la derivata temporale con uno schema di *Eulero implicito*, ottenendo il seguente problema semidiscreto:

$$\begin{aligned} &\text{noti } \mathbf{u}_h^n = \mathbf{u}_h(t^n) \in V_{h,D}(\Omega), p_h^n = p_h(t^n) \in Q_h(\Omega), \\ &\text{trovare } \mathbf{u}_h^{n+1} = \mathbf{u}_h(t^{n+1}) \in V_{h,D}(\Omega), p_h^{n+1} = p_h(t^{n+1}) \in Q_h(\Omega) : \\ &\frac{1}{\Delta t} \int_{\Omega} \mathbf{u}_h^{n+1} \cdot \mathbf{v}_h + a(\mathbf{u}_h^{n+1}, \mathbf{v}_h) + b(\mathbf{v}_h, p_h^{n+1}) = F^{n+1}(\mathbf{v}_h) + \frac{1}{\Delta t} \int_{\Omega} \mathbf{u}_h^n \cdot \mathbf{v}_h, \quad \forall \mathbf{v}_h \in V_{h,0}(\Omega), \\ & b(\mathbf{u}_h^{n+1}, q_h) = 0, \quad \forall q_h \in Q_h(\Omega), \\ & \mathbf{u}_h^0 = \mathbf{u}_{h,0}. \end{aligned}$$

Ponendo quindi

$$\mathbf{u}_h(\mathbf{x}, t) = \sum_j u_j(t) \phi_j(\mathbf{x}), \quad p_h(\mathbf{x}, t) = \sum_k p_k(t) \psi_k(\mathbf{x}),$$

la successiva discretizzazione in spazio fornisce il seguente sistema lineare

$$\begin{bmatrix} \frac{M}{\Delta t} + A & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} \mathbf{U}^{n+1} \\ \mathbf{P}^{n+1} \end{bmatrix} = \begin{bmatrix} \frac{M}{\Delta t} \mathbf{U}^n + \mathbf{F}^{n+1} \\ \mathbf{0} \end{bmatrix}, \quad (4.4)$$

dove  $M = [\int_{\Omega} \phi_j \cdot \phi_i]_{ij}$  è la matrice di massa,  $\mathbf{U}^n \in \mathbb{R}^{3 \cdot N_u}$ ,  $\mathbf{P}^n \in \mathbb{R}^{N_p}$  sono i vettori contenenti rispettivamente i gradi di libertà  $\{u_j(t^n)\}_{j=1, \dots, 3 \cdot N_u}$  e  $\{p_k(t^n)\}_{k=1, \dots, N_p}$ , e gli altri simboli mantengono lo stesso significato introdotto nel caso stazionario. La matrice del sistema lineare (4.4) è sempre la stessa ad ogni passo temporale, e dunque può essere assemblata una volta sola, all'inizio del ciclo temporale; è invece necessario aggiornare il termine noto a ciascuna iterazione. Dato  $\mathbf{U}^0$ , la risoluzione di (4.4) per ciascun  $n = 1, \dots, N$  fornisce la soluzione approssimata del problema di Stokes non stazionario ( $S_{unsteady}$ ).

### 4.3 Ottimizzazione di forma mediante *local boundary variation*

Un primo approccio computazionale per l'ottimizzazione di forma consiste nel considerare come gradi di libertà le coordinate dei nodi della griglia di calcolo appartenenti alla frontiera del corpo. Questo metodo, che prevede la variazione locale del contorno (*local boundary variation*), necessita, a partire dallo *shape gradient*  $\nabla J$ , di definire opportunamente un campo di deformazione che agisca su ciascun nodo della frontiera soggetta ad ottimizzazione: due possibili strategie vengono presentate nel seguito, insieme al trattamento del vincolo di volume.

### 4.3.1 Algoritmo di ottimizzazione numerica

Per una ampia classe di funzionali costo, il Teorema 2.20 di struttura afferma che la *shape derivative* di un funzionale costo  $J(\Omega)$  può essere espressa come

$$dJ(\Omega; \boldsymbol{\theta}) = \int_{\Gamma_c} g \boldsymbol{\theta} \cdot \mathbf{n},$$

dove  $g = g(\Omega)$  è lo *shape gradient*, dipendente dalle soluzioni del problema di stato e del problema aggiunto e  $\Gamma_c$  è la parte di frontiera soggetta a controllo ( $\boldsymbol{\theta} = \mathbf{0}$  su  $\partial\Omega \setminus \Gamma_c$ ). Questa informazione può essere sfruttata per individuare una direzione di discesa per il funzionale costo  $J(\Omega)$ : infatti, a partire da un dominio  $\Omega$ , si può ottenere un dominio deformato  $\Omega_*(\alpha)$ , con  $\alpha > 0$  parametro reale, mediante una variazione locale del contorno indotta dalla mappa perturbazione dell'identità

$$\partial\Omega_*(\alpha) = \mathbf{T}(\partial\Omega; \alpha) = (\mathbf{I} + \boldsymbol{\theta}_\alpha)(\partial\Omega),$$

scegliendo in particolare

$$\boldsymbol{\theta}_\alpha = -\alpha g(\Omega) \mathbf{n}.$$

In tal caso risulta  $J(\Omega_*(\alpha)) < J(\Omega)$ , dal momento che, per  $\alpha > 0$  sufficientemente piccolo,

$$\begin{aligned} J(\Omega_*(\alpha)) &= J(\Omega) + dJ(\Omega; \boldsymbol{\theta}_\alpha) + o(\alpha) = J(\Omega) + \int_{\Gamma_c(\Omega)} g(\Omega) \boldsymbol{\theta}_\alpha \cdot \mathbf{n} + o(\alpha) \\ &= J(\Omega) - \alpha \int_{\Gamma_c(\Omega)} g(\Omega)^2 + o(\alpha) < J(\Omega). \end{aligned}$$

Grazie alla relazione precedente, lo schema iterativo proposto nella Sezione 4.1 risulta un metodo di discesa, dove  $\Omega = \Omega^{(n)}$  (dominio al passo corrente),  $\Omega_*(\alpha) = \Omega^{(n+1)}(\alpha^{(n)})$ , ed  $\alpha = \alpha^{(n)}$  è scelto opportunamente, in modo da soddisfare, ad esempio, le seguenti condizioni di Wolfe [NW99]:

$$\begin{cases} J(\Omega^{(n+1)}(\alpha^{(n)})) & \leq J(\Omega^{(n)}) + c_1 \alpha^{(n)} dJ(\Omega^{(n)}; \boldsymbol{\theta}_{\alpha^{(n)}}), \\ dJ(\Omega^{(n+1)}(\alpha^{(n)}); \boldsymbol{\theta}_{\alpha^{(n)}}) & \geq c_2 dJ(\Omega^{(n)}; \boldsymbol{\theta}_{\alpha^{(n)}}), \end{cases}$$

con  $0 < c_1 < c_2 < 1$ .

Vi è tuttavia una ambiguità nella precedente definizione della mappa  $\mathbf{T}$ : infatti il campo di trasporto  $\boldsymbol{\theta}_\alpha$  è definito solo sul bordo di  $\partial\Omega$ . È quindi necessario estendere il campo di trasporto all'interno del dominio: da un lato, grazie al Teorema 2.20 di struttura, l'unica informazione effettivamente sfruttata per il calcolo del gradiente è il valore di  $\boldsymbol{\theta}_\alpha$  sulla frontiera  $\partial\Omega$  e dunque, per lo meno teoricamente, è possibile estendere arbitrariamente il campo di trasporto nell'interno di  $\Omega$ ; dall'altro, da un punto di vista computazionale, è opportuno trovare una estensione che restituisca un dominio deformato  $\Omega_*(\alpha)$  di regolarità sufficiente, in particolare tale da evitare costose operazioni di *remeshing*. Una semplice possibilità è quella basata sull'estensione armonica: si definisce  $\tilde{\mathbf{T}}(\mathbf{x}; \alpha) = (\mathbf{I} + \tilde{\boldsymbol{\theta}}_\alpha)(\mathbf{x})$ , dove  $\tilde{\boldsymbol{\theta}}_\alpha$  è l'estensione di  $\boldsymbol{\theta}_\alpha$  per mezzo della mappa *Dirichlet to Dirichlet*, ottenuta risolvendo il seguente problema di Laplace:

$$\begin{cases} -\Delta \tilde{\boldsymbol{\theta}}_\alpha = 0, & \text{su } \Omega, \\ \tilde{\boldsymbol{\theta}}_\alpha = 0, & \text{su } \partial\Omega \setminus \Gamma_c(\Omega), \\ \tilde{\boldsymbol{\theta}}_\alpha = \boldsymbol{\theta}_\alpha, & \text{su } \Gamma_c(\Omega). \end{cases}$$

È possibile quindi applicare la perturbazione dell'identità  $\tilde{\mathbf{T}}$  e deformare  $\Omega$  (e di conseguenza la griglia di calcolo) nel dominio  $\Omega_*(\alpha)$ , conservando il miglioramento del funzionale costo,  $J(\Omega_*(\alpha)) < J(\Omega)$ . Purtroppo questo approccio, in presenza di una scarsa regolarità di  $g(\Omega)$ , può indurre oscillazioni spurie sulla frontiera del dominio soggetto ad ottimizzazione [All07]; per ovviare a questo inconveniente sono state proposte numerose procedure di *shape regularization*, tra cui:

- utilizzare una mesh più lasca nel calcolo di  $\tilde{\boldsymbol{\theta}}_\alpha$ ;
- scegliere  $\tilde{\boldsymbol{\theta}}_\alpha$  come soluzione della seguente mappa *Dirichlet to Neumann*:

$$\begin{cases} -\Delta \tilde{\boldsymbol{\theta}}_\alpha = 0, & \text{su } \Omega, \\ \tilde{\boldsymbol{\theta}}_\alpha = 0, & \text{su } \partial\Omega \setminus \Gamma_c(\Omega), \\ \frac{\partial \tilde{\boldsymbol{\theta}}_\alpha}{\partial \mathbf{n}} = \boldsymbol{\theta}_\alpha, & \text{su } \Gamma_c(\Omega); \end{cases}$$

essa risulta, grazie ai teoremi di regolarità ellittica, più regolare rispetto alla mappa *Dirichlet to Dirichlet* e fornisce comunque una direzione di discesa per il problema di ottimizzazione di forma: infatti, ponendo  $\boldsymbol{\phi} = \tilde{\boldsymbol{\theta}}_\alpha$  nella seguente formulazione debole:

$$\int_{\Omega} \nabla \tilde{\boldsymbol{\theta}}_\alpha : \nabla \boldsymbol{\phi} = \int_{\Gamma_c(\Omega)} \boldsymbol{\theta}_\alpha \cdot \boldsymbol{\phi}, \quad \forall \boldsymbol{\phi} \in H^1_{\partial\Omega \setminus \Gamma_c(\Omega)}(\Omega),$$

si ottiene, ricordando la definizione di  $\boldsymbol{\theta}_\alpha$ ,

$$dJ(\Omega; \tilde{\boldsymbol{\theta}}_\alpha) = \int_{\Gamma_c(\Omega)} g(\Omega) \tilde{\boldsymbol{\theta}}_\alpha \cdot \mathbf{n} = -\frac{1}{\alpha} \int_{\Gamma_c(\Omega)} \boldsymbol{\theta}_\alpha \cdot \tilde{\boldsymbol{\theta}}_\alpha = -\frac{1}{\alpha} \int_{\Omega} \nabla \tilde{\boldsymbol{\theta}}_\alpha : \nabla \tilde{\boldsymbol{\theta}}_\alpha < 0.$$

### 4.3.2 Gestione del vincolo di volume

Come mostrato nella scrittura delle condizioni di ottimalità nei capitoli precedenti, il vincolo di volume  $V(\Omega) = |\Omega| = \bar{V}$  può essere imposto mediante penalizzazione, introducendo un moltiplicatore di Lagrange  $\xi$  ed ottimizzando rispetto al funzionale costo  $\tilde{J}(\Omega) = J(\Omega) + \xi V(\Omega)$ . Per quanto riguarda l'aggiornamento del moltiplicatore  $\xi^{(n)}$  nel corso delle iterazioni, si può procedere come segue:

- fissati  $\varepsilon > 0$  e  $\xi^{(0)} > 0$ , aggiornare il valore del moltiplicatore confrontando il volume della forma corrente con il volume  $\bar{V}$  di riferimento:

$$\xi^{(n+1)} = \begin{cases} \xi^{(n)} + \varepsilon, & \text{se } |\Omega^{(n)}| > \bar{V}, \\ \xi^{(n)}, & \text{se } |\Omega^{(n)}| = \bar{V}, \\ \xi^{(n)} - \varepsilon, & \text{se } |\Omega^{(n)}| < \bar{V}; \end{cases}$$

tuttavia questo tipo di aggiornamento risulta molto sensibile al valore di  $\varepsilon$  che, se scelto troppo grande, può portare ad oscillazioni sul volume dei domini della successione  $\{|\Omega^{(n)}|\}_{n \geq 0}$ ;

- aggiornare  $\xi^{(n)}$  in modo che il vincolo di volume sia verificato quando l'algoritmo converge, cioè trovare  $\xi$  tale che

$$dJ(\Omega; \boldsymbol{\theta}) + \xi dV(\Omega; \boldsymbol{\theta}) = 0,$$

nel senso della media sul contorno  $\Gamma_c$ , ovvero:

$$\xi = -\frac{\int_{\Gamma_c} g(\Omega) d\mathbf{s}}{\int_{\Gamma_c} d\mathbf{s}}.$$

Seguendo infine quanto riportato in [AP06], è possibile effettuare l'aggiornamento di  $\xi^{(n)}$  tramite la seguente relazione:

$$\xi^{(n+1)} = \frac{\xi^{(n)} + \xi}{2} + \varepsilon(|\Omega^n| - \bar{V}),$$

dove  $\varepsilon > 0$  è una costante sufficientemente piccola.

Un secondo approccio consiste nel considerare il vincolo di volume mediante proiezione, anziché mediante penalizzazione. Una possibilità, tratta da [MNPV11], è quella di considerare il dominio  $\Omega_*(\alpha)$ , ottenuto trascurando il vincolo di volume, e deformarne la frontiera, nella direzione della normale  $\mathbf{n}$ , fino quando il vincolo di volume non sia soddisfatto. Questo conduce al seguente metodo di Newton-Raphson: posto  $\Omega_0 = \Omega_*(\alpha)$ , e fissata una tolleranza  $\varepsilon > 0$ , si costruisce una successione di domini  $\Omega_i$  tali che:

$$\text{finché } \left| |\Omega_i| - \bar{V} \right| \geq \varepsilon$$

$$\Omega_{i+1} = \Omega_i - \frac{|\Omega_i| - \bar{V}}{|\partial\Omega_i|} \tilde{\mathbf{n}}(\Omega_i),$$

essendo  $\tilde{\mathbf{n}}(\Omega_i)$  una estensione della normale esterna a  $\partial\Omega_i$ . Un approccio basato su un metodo di proiezione verrà in particolare utilizzato nell'algoritmo basato su una parametrizzazione di tipo *Free Form Deformation*, illustrato nella prossima sezione.

## 4.4 Ottimizzazione di forma mediante parametrizzazione con mappa *Free Form Deformation*

In questa sezione viene presentato un metodo numerico per l'ottimizzazione di forma che sfrutta la mappa *Free Form Deformation* (*FFD*) introdotta nella Sezione 2.3.1. Questa tecnica conduce ad una notevole riduzione dimensionale rispetto all'approccio basato su *local boundary variation*, dal momento che i gradi di libertà che descrivono le deformazioni sono gli spostamenti esercitati sui punti di controllo, anziché le coordinate dei nodi di griglia situati sulla frontiera  $\Gamma_c$  da ottimizzare. Dunque, anche in presenza di mesh molto fini – richieste solitamente dalle applicazioni fluidodinamiche – il numero dei gradi di libertà non aumenta eccessivamente, come accade invece per la *local boundary variation*, ma è fissato a priori. Inoltre, una mappa di tipo *FFD* permette di ottenere una deformazione di tipo globale (applicando cioè la mappa all'intera griglia di calcolo), anziché locale come nel caso di *local boundary variation*, senza necessitare dalla risoluzione di un ulteriore problema differenziale per l'estensione del campo di spostamento  $\boldsymbol{\theta}$ . Infine, come già evidenziato nella Sezione 2.3.1, il dominio deformato gode della stessa regolarità del dominio di riferimento e dunque, a differenza della *local boundary variation*, non occorrono ulteriori operazioni di regolarizzazione della mesh deformata.

In particolare, è possibile ricavare le condizioni di ottimalità direttamente rispetto ai parametri della mappa (Sezione 4.4.1) e dunque applicare il metodo del gradiente con proiezione ai gradi di libertà  $\boldsymbol{\mu}$  (Sezione 4.4.2). Il vincolo di volume viene imposto tramite un algoritmo di tipo Newton-Raphson, descritto nella Sezione 4.4.3. Infine, l'implementazione di una libreria per la risoluzione di problemi di ottimizzazione di forma è brevemente descritta nella Sezione 4.4.4, rimandando all'Appendice A per ulteriori dettagli.

### 4.4.1 Condizioni di ottimalità rispetto ai parametri della mappa *Free Form Deformation*

Seguendo il formalismo introdotto da C ea in [C ea86], e definendo una opportuna trasformazione dei domini,   possibile calcolare il gradiente del funzionale costo  $J(\Omega)$  rispetto ai parametri  $\boldsymbol{\mu}_{k,l,m}$  della mappa *FFD*.

Siano  $\Omega$  la configurazione di riferimento su cui   definita la mappa *FFD* e  $\Omega_{\boldsymbol{\mu}} = \Omega_o(\boldsymbol{\mu})$  un dominio deformato; considerando il dominio perturbato  $\Omega_{\boldsymbol{\mu}+t\boldsymbol{\eta}} = \Omega_o(\boldsymbol{\mu} + t\boldsymbol{\eta})$ , dove

$$\begin{aligned} \mathbf{x}_{\boldsymbol{\mu}} \in \Omega_{\boldsymbol{\mu}} : & \quad \mathbf{x}_{\boldsymbol{\mu}} = \mathbf{T}(\mathbf{x}; \boldsymbol{\mu}), \\ \mathbf{x}_{\boldsymbol{\mu}+t\boldsymbol{\eta}} \in \Omega_{\boldsymbol{\mu}+t\boldsymbol{\eta}} : & \quad \mathbf{x}_{\boldsymbol{\mu}+t\boldsymbol{\eta}} = \mathbf{T}(\mathbf{x}; \boldsymbol{\mu} + t\boldsymbol{\eta}), \end{aligned}$$

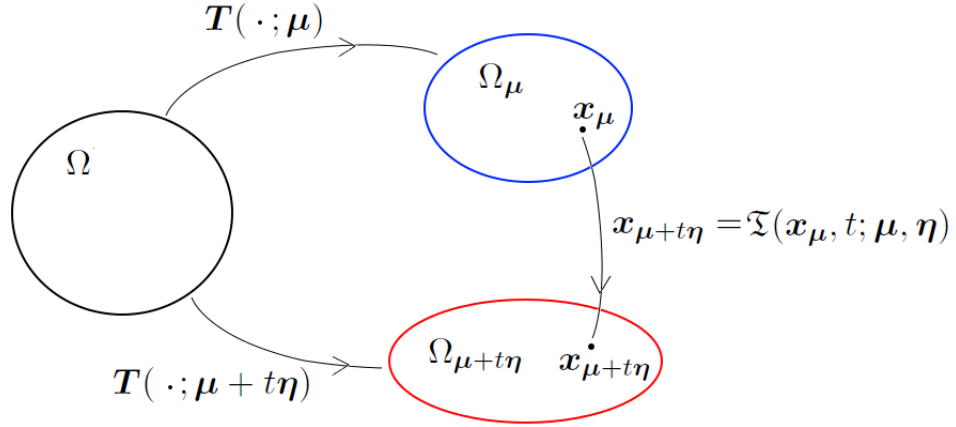


Figura 4.1: Diagramma della costruzione della mappa  $\mathfrak{T}(\mathbf{x}_\mu, t; \boldsymbol{\mu}, \boldsymbol{\eta})$  associata alla *Free Form Deformation*. Per semplificare la rappresentazione grafica si considera  $\Omega \subset \mathbb{R}^2$ .

è possibile definire la seguente trasformazione (formalmente):

$$\mathfrak{T} : \Omega_\mu \rightarrow \Omega_{\mu+t\eta}, \quad \mathbf{x}_\mu \mapsto \mathbf{x}_{\mu+t\eta},$$

data da

$$\mathfrak{T}(\mathbf{x}_\mu, t; \boldsymbol{\mu}, \boldsymbol{\eta}) = \mathbf{T}(\mathbf{T}^{-1}(\mathbf{x}_\mu; \boldsymbol{\mu}); \boldsymbol{\mu} + t\boldsymbol{\eta}).$$

Il campo di trasporto  $\boldsymbol{\theta}_\mu = \boldsymbol{\theta}_\mu(\mathbf{x}_\mu)$  definito come

$$\boldsymbol{\theta}_\mu(\mathbf{x}_\mu) = \partial_t \mathfrak{T}(\mathbf{x}_\mu, 0; \boldsymbol{\mu}, \boldsymbol{\eta}) = \sum_{k,l,m} b_{k,l,m}^{K,L,M}(\boldsymbol{\psi}(\mathbf{T}^{-1}(\mathbf{x}_\mu; \boldsymbol{\mu}))) \mathfrak{D}\boldsymbol{\eta}_{k,l,m},$$

permette di ottenere  $\Omega_{\mu+t\eta}$  applicando una perturbazione dell'identità a  $\Omega_\mu$ .

Ricordando che, per una ampia classe di funzionali costo, il Teorema 2.20 di struttura afferma che

$$dJ(\Omega_\mu; \boldsymbol{\theta}_\mu) = \int_{\partial\Omega_\mu} g \boldsymbol{\theta}_\mu \cdot \mathbf{n},$$

sostituendo l'espressione del campo di spostamento  $\boldsymbol{\theta}_\mu$  si ottiene:

$$\begin{aligned} dJ(\Omega_\mu; \boldsymbol{\theta}_\mu) &= \sum_{k,l,m} \int_{\partial\Omega_\mu} b_{k,l,m}^{K,L,M}(\boldsymbol{\psi}(\mathbf{T}^{-1}(\mathbf{x}_\mu; \boldsymbol{\mu}))) \mathfrak{D}\boldsymbol{\eta}_{k,l,m} \cdot \mathbf{n} g(\mathbf{x}_\mu) = \\ &= \sum_{k,l,m} \int_{\Omega_\mu} \operatorname{div} \left[ b_{k,l,m}^{K,L,M}(\boldsymbol{\psi}(\mathbf{T}^{-1}(\mathbf{x}_\mu; \boldsymbol{\mu}))) \mathfrak{D}\boldsymbol{\eta}_{k,l,m} g(\mathbf{x}_\mu) \right] = \\ &= \sum_{k,l,m} \boldsymbol{\eta}_{k,l,m} \cdot \int_{\Omega_\mu} \mathfrak{D}\nabla \left[ b_{k,l,m}^{K,L,M}(\boldsymbol{\psi}(\mathbf{T}^{-1}(\mathbf{x}_\mu; \boldsymbol{\mu}))) g(\mathbf{x}_\mu) \right] = \\ &= \sum_{k,l,m} \boldsymbol{\eta}_{k,l,m} \cdot \nabla_{\boldsymbol{\eta}_{k,l,m}} J(\Omega_\mu), \end{aligned} \tag{4.5}$$

avendo posto

$$\begin{aligned} \nabla_{\boldsymbol{\eta}_{k,l,m}} J(\Omega_\mu) &= \int_{\Omega_\mu} \mathfrak{D}\nabla \left[ b_{k,l,m}^{K,L,M}(\boldsymbol{\psi}(\mathbf{T}^{-1}(\mathbf{x}_\mu; \boldsymbol{\mu}))) g(\mathbf{x}_\mu) \right] \\ &= \int_{\Omega_\mu} \mathfrak{D}\nabla g(\mathbf{x}_\mu) b_{k,l,m}^{K,L,M}(\boldsymbol{\psi}(\mathbf{T}^{-1}(\mathbf{x}_\mu; \boldsymbol{\mu}))) + \\ &+ \int_{\Omega_\mu} \mathfrak{D}\nabla \mathbf{T}^{-T}(\mathbf{x}_\mu; \boldsymbol{\mu}) \mathfrak{D}^{-1} \nabla b_{k,l,m}^{K,L,M}(\boldsymbol{\psi}(\mathbf{T}^{-1}(\mathbf{x}_\mu; \boldsymbol{\mu}))) g(\mathbf{x}_\mu). \end{aligned} \tag{4.6}$$



Il passaggio dalla prima alla seconda riga di (4.5) è giustificato dal Teorema della divergenza, mentre l'uguaglianza dalla seconda riga alla terza si ricava tramite l'identità vettoriale

$$\operatorname{div}(s\mathbf{V}) = s \operatorname{div} \mathbf{V} + \mathbf{V} \cdot \nabla s$$

valida per ogni  $s : \mathbb{R}^3 \rightarrow \mathbb{R}$  e per ogni  $\mathbf{V} : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ . La (4.6) evidenzia che, supponendo lo *shape gradient*  $g$  sufficientemente regolare, il gradiente del funzionale costo rispetto a  $\boldsymbol{\eta}_{k,l,m}$  si può ricavare integrando opportune funzioni dipendenti in particolare da  $g$ ,  $\nabla g$  e dagli indici  $k, l, m$ .

Indicando con  $\nabla_{\boldsymbol{\eta}} J(\Omega_{\boldsymbol{\mu}})$  il vettore contenente le componenti di  $\{\nabla_{\boldsymbol{\eta}_{k,l,m}} J(\Omega_{\boldsymbol{\mu}})\}_{k,l,m}$  associate ai gradi di libertà della *FFD*, è possibile eseguire la minimizzazione del funzionale costo  $J$  ottimizzando rispetto a ciascuna componente del vettore  $\boldsymbol{\mu}$  mediante un metodo di tipo gradiente, come discusso nella prossima sezione.

#### 4.4.2 Algoritmo di ottimizzazione numerica

Grazie alla scrittura della condizione di ottimalità rispetto ai parametri  $\boldsymbol{\mu}$  della mappa *Free Form Deformation* descritta in precedenza, il problema di ottimizzazione di forma è ricondotto al seguente problema di ottimizzazione vincolata

$$\min_{\boldsymbol{\mu} \in M_{ad}} j(\boldsymbol{\mu}), \quad (4.7)$$

essendo  $j(\boldsymbol{\mu}) = J(\Omega_{\boldsymbol{\mu}})$ ,  $M_{ad} \subset M = \mathbb{R}^p$ , dove  $p$  indica il numero dei gradi di libertà. L'insieme  $M_{ad}$  può tenere conto sia dei vincoli fisici imposti sulle forme ammissibili (in particolare, un vincolo di volume), sia di vincoli dettati dall'implementazione della *FFD* (ad esempio, un vincolo che i gradi di libertà contenuti in  $\boldsymbol{\mu}$  siano scelti sufficientemente piccoli da soddisfare le ipotesi del Teorema 2.15). Supponendo  $j \in C^1(\mathbb{R}^p; \mathbb{R})$  e  $M_{ad} \subset M$  insieme convesso, chiuso e non vuoto, numerosi metodi sono stati proposti per la risoluzione di un generico problema di ottimizzazione vincolata (4.7): uno tra i più semplici è quello del gradiente con proiezione (*Gradient Projection*), ideato da Goldstein [Gol64] e Levitin-Polyak [LP66].

In base a questo metodo, a partire da un valore iniziale  $\boldsymbol{\mu}^{(0)} \in M_{ad}$ , si costruisce una successione  $\{\boldsymbol{\mu}^{(n)}\}_{n \in \mathbb{N}}$  tale che

$$\boldsymbol{\mu}^{(n+1)} = \boldsymbol{\mu}^{(n+1)}(\alpha^{(n)}) = \mathbf{P}(\boldsymbol{\mu}^{(n+1)} - \alpha^{(n)} \nabla j(\boldsymbol{\mu}^{(n)})), \quad (4.8)$$

dove  $\mathbf{P}$  denota l'operatore di proiezione su  $M_{ad}$ ; la convergenza risulta lineare se  $\alpha^{(n)}$  è una costante sufficientemente piccola e  $\nabla j$  è una funzione Lipschitziana. Nel caso particolare in cui  $M_{ad} = M$  si ritrova il classico metodo del gradiente per l'ottimizzazione non vincolata, in cui la successione  $\{\boldsymbol{\mu}^{(n)}\}_{n \in \mathbb{N}}$  è costruita come

$$\boldsymbol{\mu}^{(n+1)} = \boldsymbol{\mu}^{(n+1)}(\alpha^{(n)}) = \boldsymbol{\mu}^{(n)} - \alpha^{(n)} \nabla j(\boldsymbol{\mu}^{(n)}).$$

Inoltre, per quanto riguarda la scelta del passo  $\alpha^{(n)}$ , la regola di Armijo può essere generalizzata nel modo seguente [Ber76]: dati  $\boldsymbol{\mu}^{(n)} \in M_{ad}$ ,  $\bar{\alpha} > 0$ , un fattore di contrazione  $\rho \in (0, 1)$  e  $c \in (0, 1]$ , si può scegliere  $\alpha^{(n)}$  con un metodo di *backtracking*, ovvero come

$$\alpha^{(n)} = \rho^{m_n} \bar{\alpha}, \quad (4.9)$$

per il più piccolo  $m_n \in \mathbb{N}$  tale che

$$j(\boldsymbol{\mu}^{(n+1)}(\alpha^{(n)})) \leq j(\boldsymbol{\mu}^{(n)}) + c \nabla j(\boldsymbol{\mu}^{(n)}) \cdot (\boldsymbol{\mu}^{(n+1)}(\alpha^{(n)}) - \boldsymbol{\mu}^{(n)}). \quad (4.10)$$

Infatti, se  $M_{ad} = M$ , la (4.10) si riduce a

$$j(\boldsymbol{\mu}^{(n+1)}(\alpha^{(n)})) \leq f(\boldsymbol{\mu}^{(n)}) - c \alpha^{(n)} \nabla j(\boldsymbol{\mu}^{(n)}) \cdot \nabla j(\boldsymbol{\mu}^{(n)}),$$

che è la regola di Armijo per il metodo del gradiente.

Lo svantaggio del metodo del gradiente con proiezione è la necessità di una formulazione esplicita dell'operatore  $\mathbf{P}$ : essa risulta solitamente disponibile solo per vincoli semplici (si veda ad esempio [NW99]), come i seguenti:

- se  $M_{ad} = M_{ad}^b = \prod_{i=1}^p [a_i, b_i]$ , allora l'operatore di proiezione è

$$\mathbf{P}_b(\boldsymbol{\mu}) = (\min(\max(a_i, \mu_i), b_i))_{i=1, \dots, p}. \quad (4.11)$$

In un problema di ottimizzazione di forma è necessario imporre un vincolo di questo tipo per garantire, ad esempio, che i gradi di libertà siano scelti sufficientemente piccoli in modo da soddisfare le ipotesi del Teorema 2.15.

- se  $M_{ad} = M_{ad}^l = \{\boldsymbol{\mu} \in M : A\boldsymbol{\mu} = \mathbf{b}\}$ , dove  $A \in \mathbb{R}^{q \times p}$  ha rango  $q \ll p$ ,  $\mathbf{b} \in \mathbb{R}^q$ , allora l'operatore di proiezione<sup>1</sup> è

$$\mathbf{P}_l(\boldsymbol{\mu}) = \boldsymbol{\mu} - A^T(AA^T)^{-1}(A\boldsymbol{\mu} - \mathbf{b}). \quad (4.12)$$

In un problema di ottimizzazione di forma è necessario imporre un vincolo di questo tipo per garantire, ad esempio, che un punto  $\bar{\mathbf{x}}$  rimanga fisso nella posizione che occupa nella configurazione di riferimento: infatti, grazie all'espressione (2.17) della mappa *FFD* come perturbazione dell'identità, ciò è equivalente ad imporre il vincolo  $\boldsymbol{\theta}(\bar{\mathbf{x}}; \boldsymbol{\mu}) = \mathbf{0}$  che, in virtù dell'espressione (2.18) per il campo di spostamento  $\boldsymbol{\theta}(\mathbf{x}; \boldsymbol{\mu})$ , assume la struttura di un vincolo lineare in  $\boldsymbol{\mu}$ .

Non è tuttavia possibile derivare agevolmente l'espressione dell'operatore di proiezione in casi più complessi, come ad esempio quello in cui il vettore dei gradi di libertà debba soddisfare contemporaneamente sia vincoli di *bound* sia vincoli lineari, cioè  $M_{ad} = M_{ad}^{b \cap l} = \{\boldsymbol{\mu} \in \prod_{i=1}^p [a_i, b_i] : A\boldsymbol{\mu} = \mathbf{b}\}$ : si può, comunque, pensare di applicare in modo alterno i due operatori  $\mathbf{P}_b$  e  $\mathbf{P}_l$  fino a convergenza<sup>2</sup>.

### 4.4.3 Gestione del vincolo di volume

Come già discusso, un vincolo molto comune nei problemi di ottimizzazione di forma riguarda il volume del corpo soggetto ad ottimizzazione: assegnato  $\bar{V} > 0$ , è necessario deformare il dominio in modo che

$$V(\Omega) = \bar{V} \quad \forall \Omega \text{ ammissibile.}$$

Nel caso della parametrizzazione mediante mappa *FFD*, è quindi possibile definire l'insieme  $M_{ad}$  degli spostamenti che soddisfano il vincolo di volume come

$$M_{ad} = \{\boldsymbol{\mu} \in M : v(\boldsymbol{\mu}) = 0\}, \quad \text{essendo } v(\boldsymbol{\mu}) = V(\Omega_{\boldsymbol{\mu}}) - \bar{V}.$$

Si può mostrare che  $v(\boldsymbol{\mu}) = 0$  è un vincolo non lineare; con l'obiettivo di eseguirne una linearizzazione, si ricava l'espressione di  $\nabla_{\boldsymbol{\eta}} v(\boldsymbol{\mu}) = \nabla_{\boldsymbol{\eta}} V(\Omega_{\boldsymbol{\mu}})$ : per una generica mappa *perturbazione dell'identità*  $\mathbf{T} = \mathbf{I} + \boldsymbol{\theta}$ , la derivata di  $V(\Omega)$  rispetto alla forma è data da

$$dV(\Omega; \boldsymbol{\theta}) = \int_{\partial\Omega} \boldsymbol{\theta} \cdot \mathbf{n};$$

la formulazione di  $\nabla_{\boldsymbol{\eta}_{k,l,m}} V(\Omega_{\boldsymbol{\mu}})$  può essere dunque ottenuta da (4.6) ponendo  $g(\Omega) = 1$  ed, infine, è possibile ricavare il gradiente del vincolo di volume  $\nabla_{\boldsymbol{\eta}} v(\Omega_{\boldsymbol{\mu}})$  considerando nell'insieme  $\{\nabla_{\boldsymbol{\eta}_{k,l,m}} V(\Omega_{\boldsymbol{\mu}})\}_{k,l,m}$  solo le componenti associate a spostamenti abilitati.

Per risolvere il problema di ottimizzazione vincolata

$$\min_{\boldsymbol{\mu} \in M_{ad}} j(\boldsymbol{\mu}) \quad (4.13)$$

<sup>1</sup>Basterebbe  $q \leq p$ . La richiesta  $q \ll p$  è solo per garantire ipotesi favorevoli in cui l'inversione esplicita della matrice  $AA^T$  sia poco onerosa.

<sup>2</sup>La successione così ottenuta è convergente perché la norma di un operatore di proiezione è minore o uguale a 1.

mediante il metodo del gradiente con proiezione presentato nella sezione precedente, è necessario fornire l'approssimazione di una<sup>3</sup> proiezione  $Q(\boldsymbol{\mu})$  di  $\boldsymbol{\mu} \in M$  su  $M_{ad}$ .

Infatti, più in generale, supponendo assegnata una funzione  $\mathbf{v} : \mathbb{R}^p \rightarrow \mathbb{R}^q$  tale che  $\nabla \mathbf{v}(\boldsymbol{\mu})$  abbia rango  $q \ll p$  per ogni  $\boldsymbol{\mu} \in M$ , una proiezione di  $\bar{\boldsymbol{\mu}} \in M$  su  $M_{ad} = \{\boldsymbol{\mu} \in M : \mathbf{v}(\boldsymbol{\mu}) = \mathbf{0}\}$  è data da

$$Q(\bar{\boldsymbol{\mu}}) = \bar{\boldsymbol{\mu}} + \boldsymbol{\delta}(\bar{\boldsymbol{\mu}}),$$

dove  $\boldsymbol{\delta}(\bar{\boldsymbol{\mu}})$  è una soluzione del seguente problema di ottimizzazione

$$\begin{aligned} \min \|\boldsymbol{\delta}(\bar{\boldsymbol{\mu}})\|^2 \\ \text{s.t. } \mathbf{v}(\bar{\boldsymbol{\mu}} + \boldsymbol{\delta}(\bar{\boldsymbol{\mu}})) = \mathbf{0}. \end{aligned}$$

Dato che, nello schema iterativo della (4.8), è necessario risolvere una istanza di questo problema per ciascun  $n$ , con  $\bar{\boldsymbol{\mu}} = \boldsymbol{\mu}^{(n+1)} - \alpha^{(n)} \nabla j(\boldsymbol{\mu}^{(n)})$ , e poiché la risoluzione esatta di questo problema è eccessivamente onerosa, si considera una risoluzione approssimata con un algoritmo di tipo Newton-Raphson. Posto  $\mathbf{q}_0 = \bar{\boldsymbol{\mu}}$ , e fissata una tolleranza  $\varepsilon > 0$ , finché  $\|\mathbf{v}(\mathbf{q}_i)\| \geq \varepsilon$  si risolve

$$\begin{aligned} \min \|\boldsymbol{\delta}_i\|^2 \\ \text{s.t. } \mathbf{v}(\mathbf{q}_i) + \nabla \mathbf{v}(\mathbf{q}_i) \boldsymbol{\delta}_i = \mathbf{0}, \end{aligned} \quad (4.14)$$

e si aggiorna la variabile  $\mathbf{q}_{i+1} = \mathbf{q}_i + \boldsymbol{\delta}_i$ .

È possibile ricavare esplicitamente il vettore  $\boldsymbol{\delta}_i$ , a ciascuna iterazione, mediante la (4.12): infatti, la (4.14) è la definizione della proiezione del vettore  $\mathbf{0}$  sul sottospazio affine  $\{\boldsymbol{\delta} \in M : \nabla \mathbf{v}(\mathbf{q}_i) \boldsymbol{\delta} = -\mathbf{v}(\mathbf{q}_i)\}$ ; si ottiene quindi che la formula di aggiornamento per  $\mathbf{q}_{i+1}$  è data da

$$\mathbf{q}_{i+1} = \mathbf{q}_i - \nabla \mathbf{v}(\mathbf{q}_i)^T (\nabla \mathbf{v}(\mathbf{q}_i) \nabla \mathbf{v}(\mathbf{q}_i)^T)^{-1} \mathbf{v}(\mathbf{q}_i), \quad (4.15)$$

ed, indicando con  $I$  il primo indice tale che  $\|\mathbf{v}(\mathbf{q}_I)\| < \varepsilon$ , si pone infine  $Q(\bar{\boldsymbol{\mu}}) \approx Q_\varepsilon(\bar{\boldsymbol{\mu}}) = \mathbf{q}_I$ :  $Q_\varepsilon(\bar{\boldsymbol{\mu}})$  soddisfa il vincolo a meno di una tolleranza  $\varepsilon$  ed è un'approssimazione di una proiezione di  $\bar{\boldsymbol{\mu}}$  su  $M_{ad}$ , perché ciascun incremento  $\boldsymbol{\delta}_i$  è stato scelto effettuando una proiezione sul vincolo linearizzato.

#### 4.4.4 Presentazione della libreria C++

L'implementazione in C++ della libreria per la risoluzione approssimata di un problema di ottimizzazione di forma si basa sostanzialmente su i tre "ingredienti" seguenti:

1. una *parametrizzazione della forma*: la parametrizzazione mediante mappa *Free Form Deformation* è stata implementata nelle classi del *namespace* `ShapeParametrization`, presentate in dettaglio nell'Appendice A.1;
2. un *risolutore ad elementi finiti* per il problema di stato e per il problema aggiunto, fornito dalla libreria *LifeV* (versione parallela);
3. un *metodo di ottimizzazione*: la risoluzione di un generico problema di ottimizzazione vincolata mediante il metodo del gradiente con proiezione è stata implementata nelle classi del *namespace* `NumericalOptimization`, presentate in dettaglio nell'Appendice A.2.

Questi ingredienti vengono interfacciati nel *namespace* `ShapeOptimizationProblem`, presentato nella Appendice A.3, ed in particolare nella classe `FFDProjectionMethodShapeOptimizationProblem`, che permette di risolvere un generico problema di *shape optimization* per la famiglia di perturbazioni dell'identità fornita dalla *FFD*, in cui l'ottimizzazione rispetto ai parametri  $\boldsymbol{\mu}$  è effettuata mediante del metodo del gradiente con proiezione. La libreria che si ottiene in questo modo può essere applicata ad una vasta classe di problemi di ottimizzazione di forma bidimensionali e tridimensionali, stazionari e non stazionari. Infatti:

<sup>3</sup>La proiezione potrebbe non essere unica perché  $M_{ad}$  non è necessariamente convesso.

- grazie agli strumenti messi a disposizione dalla libreria *Life V*, è possibile importare la mesh del dominio di riferimento su cui applicare la *FFD* secondo i formati disponibili nei più comuni generatori (*Netgen*, *Gmsh*, formato *mesh++*, ...). Durante il ciclo di ottimizzazione non è invece mai necessario interfacciarsi con questi generatori, dal momento che non viene mai effettuata alcuna operazione di *remeshing*: tutti i domini sono infatti ottenuti per deformazione mediante mappa *FFD* del dominio di riferimento, e presentano un grado elevato di regolarità grazie alla struttura della mappa;
- la scelta dei gradi di libertà della mappa *FFD* non è fissata a priori, ma è lasciata all'utente: in base alle necessità del problema in esame, si può quindi optare per una griglia di punti di controllo più o meno fine, selezionando i gradi di libertà ritenuti più significativi per l'applicazione;
- si può imporre qualunque tipo di vincolo sulle forme ammissibili: in particolare, sono già disponibili formulazioni per i vincoli più comuni (ad esempio il vincolo di volume);
- è possibile considerare qualunque problema di stato tra quelli disponibili in *Life V*, stazionario o non stazionario, bidimensionale o tridimensionale;
- è possibile considerare qualunque funzionale costo del tipo

$$J(\Omega) = \int_{\Omega} j_{\Omega}(\mathbf{x}, u(\Omega), \nabla u(\Omega)) d\mathbf{x},$$

per un problema di stato stazionario, oppure

$$J(\Omega) = \int_0^T \int_{\Omega} j_{\Omega}(\mathbf{x}, t, u(\Omega, t), \nabla u(\Omega, t)) d\mathbf{x} dt + \int_{\Omega} j_{\Omega}^T(\mathbf{x}, T, u(\Omega, T), \nabla u(\Omega, T)) d\mathbf{x},$$

per un problema di stato non stazionario;

- la libreria è predisposta per la risoluzione del problema di ottimizzazione rispetto ai parametri  $\boldsymbol{\mu}$  della mappa *FFD* mediante il metodo del gradiente con proiezione: tuttavia, grazie alla struttura gerarchica presentata nell'Appendice A.3, è possibile, con sufficiente facilità, cambiare il metodo di ottimizzazione ed utilizzare librerie esterne che implementino algoritmi più efficienti (ad esempio metodi delle direzioni coniugate o metodi di tipo quasi-Newton).

Nel prossimo capitolo verranno presentati e commentati i risultati ottenuti risolvendo alcuni problemi di ottimizzazione di forma mediante questa libreria.

## Capitolo 5

# Simulazioni numeriche

In questo capitolo vengono illustrati i risultati delle simulazioni numeriche relative a sei problemi di ottimizzazione di forma, ottenuti sfruttando la tecnica di *Free Form Deformation* e la libreria C++ presentata nella Sezione 4.4.4 e nell'Appendice A. La prima applicazione (Sezione 5.1) riguarda un caso test stazionario per l'equazione di Poisson ed un caso test non stazionario per l'equazione del calore, per i quali, in entrambi i casi, è disponibile analiticamente una forma ottima. La seconda applicazione (Sezione 5.2), prevede l'ottimizzazione della forma di un corpo immerso in un flusso di Stokes, stazionario e non stazionario: il problema è stato ampiamente studiato in passato e per il caso stazionario è disponibile analiticamente la forma ottima. Infine, la Sezione 5.3 presenta i risultati ottenuti per l'ottimizzazione della forma di un bypass aorto-coronarico, secondo le ipotesi introdotte nella Sezione 1.3. I risultati del modello di Stokes stazionario sono descritti nella Sezione 5.3.1; il problema non stazionario, fisiologicamente più realistico, è illustrato nella Sezione 5.3.2. Un confronto delle configurazioni ottime ottenute mediante un approccio di tipo *Free Form Deformation* con alcuni risultati di studi precedenti è infine presentato nella Sezione 5.3.3.

### 5.1 Due casi test con equazione di Poisson

In questa sezione vengono presentati due casi test (uno stazionario ed uno non stazionario), per i quali è disponibile analiticamente la forma ottima.

#### 5.1.1 Caso test stazionario

Si consideri il seguente problema stazionario: dati  $f \in L^2(\mathbb{R}^n)$ ,  $g \in H^2(\mathbb{R}^n)$ , sia  $u = u(\Omega) \in H^1(\Omega)$  soluzione del problema

$$\begin{cases} -\Delta u(\Omega) = f, & \text{in } \Omega, \\ u(\Omega) = g, & \text{su } \partial\Omega, \end{cases}$$

e, assegnato  $\hat{y} \in L^2(\mathbb{R}^n)$ , si minimizzi il seguente funzionale costo:

$$J(\Omega) = \int_{\Omega} |u(\Omega) - \hat{u}|^2.$$

L'analisi del problema di ottimizzazione di forma

$$\hat{\Omega} = \arg \min_{\Omega \subset \mathbb{R}^n} J(\Omega), \tag{5.1}$$

#	Ordine FE	Punti di controllo	Vincolo di volume
1	1	sfera	no
2	1	sfera	sì
3	2	sfera	no
4	2	sfera	sì
5	2	cubo	sì

#	Numero di iterazioni	$J^{(n)}$	$ J^{(n)} - J^{(n-1)} $	$\ \nabla_{\boldsymbol{\mu}} J^{(n)}\ _2$	$\ \boldsymbol{\mu}^{(n)}\ _2$
1	10	$3.948 \times 10^0$	$*6.262 \times 10^{-5}$	$1.424 \times 10^{-1}$	$1.215 \times 10^0$
2	4	$3.315 \times 10^{-1}$	$7.507 \times 10^{-2}$	$2.053 \times 10^{-2}$	$5.657 \times 10^{-1}$
3	16	$6.560 \times 10^{-4}$	$*7.726 \times 10^{-5}$	$1.013 \times 10^{-3}$	$7.027 \times 10^{-2}$
4	8	$6.761 \times 10^{-4}$	$1.305 \times 10^{-4}$	$*9.911 \times 10^{-4}$	$7.092 \times 10^{-2}$
5	5	$1.677 \times 10^{-2}$	$6.063 \times 10^{-3}$	$*8.954 \times 10^{-4}$	$5.442 \times 10^{-1}$

Tabella 5.1: Risultati del caso test stazionario. La simulazione viene interrotta quando  $|J^{(n)} - J^{(n-1)}| < 10^{-4}$  oppure  $\|\nabla_{\boldsymbol{\mu}} J^{(n-1)}\|_2 < 10^{-3}$ : il criterio di arresto che ha causato la terminazione della simulazione è indicato con un asterisco. La simulazione 2 è stata arrestata da un'eccezione nel ciclo di *backtracking* (impossibilità di trovare un passo  $\alpha_4$  tale che la condizione di Armijo sia verificata). La configurazione dei punti di controllo denominata con 'sfera' si riferisce alla Figura 5.1(a), quella denominata 'cubo' alla Figura 5.1(b).

è già stata affrontata nel Capitolo 2; in particolare le condizioni di ottimalità, ricavate per mezzo di due differenti approcci nelle Sezioni 2.4.1 e 2.4.2, sono:

$$\begin{cases} -\Delta u(\Omega) = f, & \text{in } \Omega, \\ u(\Omega) = g, & \text{su } \partial\Omega, \end{cases} \quad \text{problema di stato,}$$

$$\begin{cases} -\Delta p(\Omega) = -2(u(\Omega) - \hat{u}), & \text{in } \Omega, \\ p(\Omega) = 0, & \text{su } \partial\Omega, \end{cases} \quad \text{problema aggiunto,}$$

$$dJ(\Omega; \boldsymbol{\theta}) = \int_{\partial\Omega} \left\{ |u(\Omega) - \hat{u}|^2 - \frac{\partial}{\partial \mathbf{n}} p(\Omega) \frac{\partial}{\partial \mathbf{n}} (u(\Omega) - g) \right\} \boldsymbol{\theta} \cdot \mathbf{n}, \quad \text{equazione di sensitività.}$$

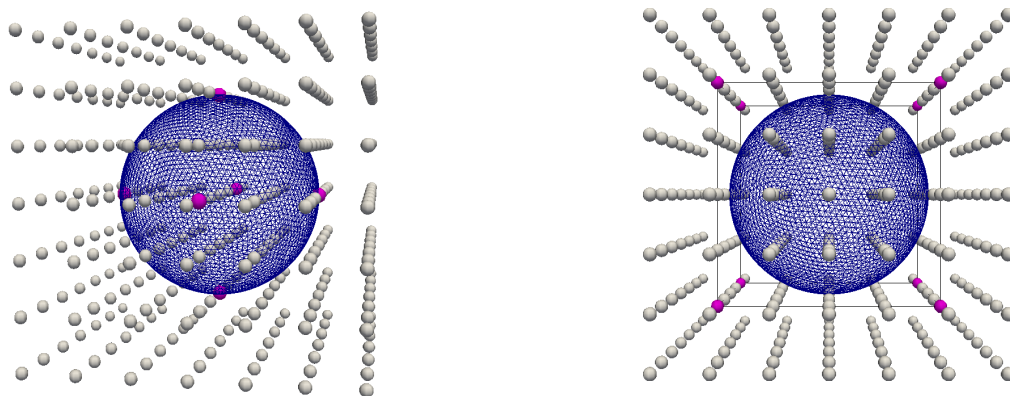
Nel seguito si pone  $n = 3$ ,  $f(x_1, x_2, x_3) = 0$ ,  $g(x_1, x_2, x_3) = x_1^2 + 2x_2^2 + 3x_3^2$  ed  $\hat{u} = u(B_2(\mathbf{0}))$ , cioè

$$\begin{cases} -\Delta \hat{u} = f, & \text{in } \hat{\Omega} = B_{r=2}(\mathbf{x} = \mathbf{0}), \\ \hat{u} = g, & \text{su } \partial\hat{\Omega}. \end{cases} \quad (5.2)$$

Il sistema (5.2) può essere risolto per separazione delle variabili in coordinate sferiche, trovando  $\hat{u}(x_1, x_2, x_3) = x_3^2 - x_1^2 + 8$ . Il problema (5.1) ammette quindi una forma ottima data da  $\hat{\Omega} = B_2(\mathbf{0})$ , per la quale il funzionale costo è pari a  $J(\hat{\Omega}) = 0$ .

Si desidera dunque approssimare la forma mediante parametrizzazione *FFD* e ricercare i parametri  $\boldsymbol{\mu}$  ottimi con il metodo di discesa del gradiente con proiezione, utilizzando la libreria presentata nella Sezione 4.4.4 e nell'Appendice A.

Le simulazioni sono state eseguite utilizzando due diverse griglie di punti di controllo, mostrate in Figura 5.1; sono stati considerati sia elementi finiti  $\mathbb{P}^1$ , sia elementi finiti  $\mathbb{P}^2$ . Nel primo caso la



(a) 6 punti di controllo abilitati, agli estremi dei tre diametri paralleli agli assi coordinati

(b) 8 punti di controllo abilitati, ai vertici di un cubo con spigoli di lunghezza pari al diametro e paralleli agli assi coordinati

Figura 5.1: Configurazione dei punti di controllo. I punti in grigio sono vincolati, quelli in rosa hanno tre gradi di libertà.

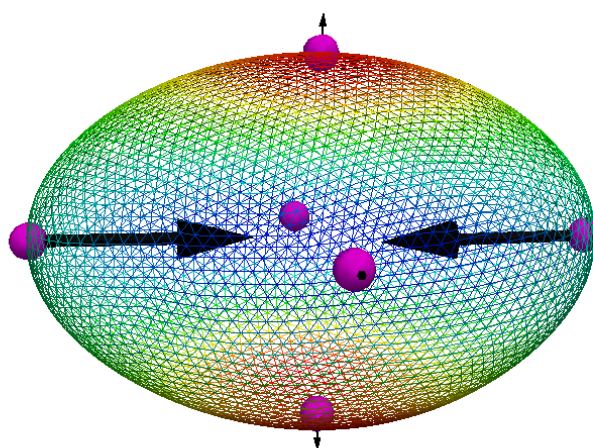
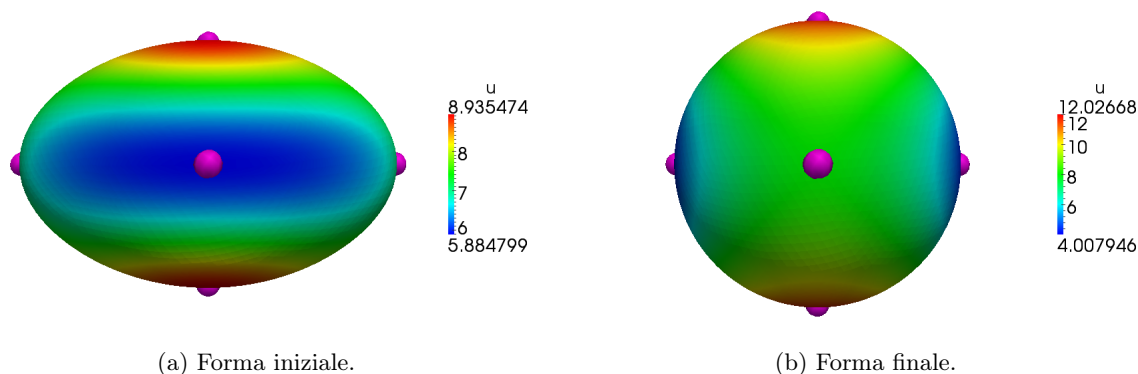


Figura 5.2: Direzioni di discesa all'iterazione iniziale (simulazione 4).



(a) Forma iniziale.

(b) Forma finale.

Figura 5.3: Forma iniziale e finale per la simulazione 4.

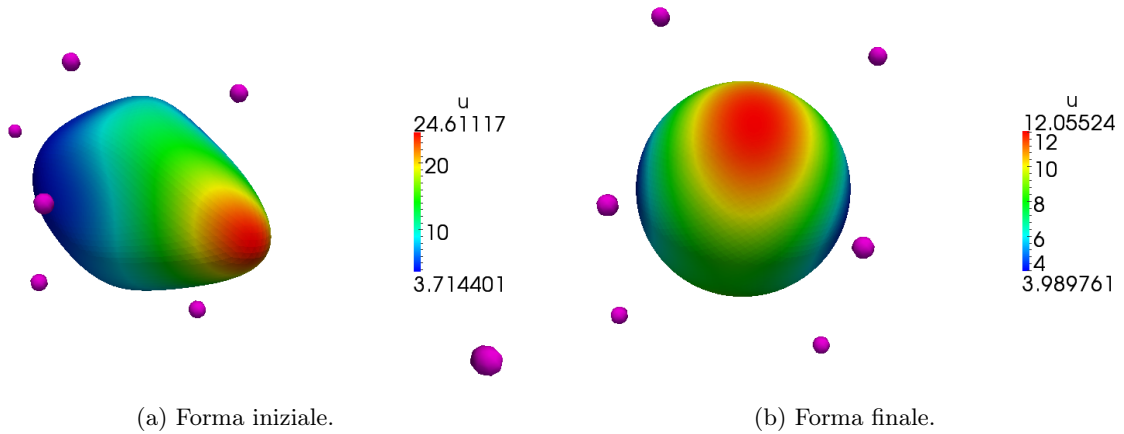


Figura 5.4: Forma iniziale e finale per la simulazione 5.

simulazione è stata eseguita sul server *dougga*<sup>1</sup> con 2 processori, nel secondo sul cluster *idra*<sup>2</sup> con 4 oppure 8 processori. Si è scelto inoltre di imporre, in alcune simulazioni, il vincolo di volume  $V(\Omega) = V(\hat{\Omega})$ .

La configurazione di riferimento  $\Omega$ , coincidente con la forma ottima  $\hat{\Omega}$ , è stata triangolata con circa 90.000 elementi di volume e deformata imponendo un vettore di spostamenti  $\boldsymbol{\mu}$  non nullo, scelto in modo che il funzionale costo all'iterazione iniziale sia dell'ordine di  $10^2$ : la forma ottima si ottiene quindi per  $\boldsymbol{\mu} \approx \mathbf{0}$ .

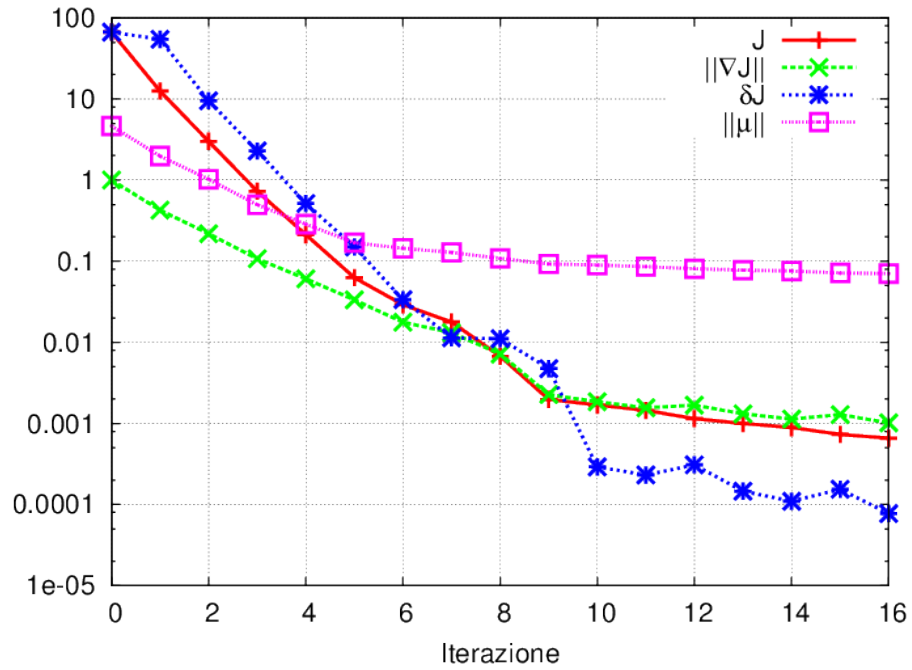
I risultati riassunti in Tabella 5.1 mettono in evidenza quanto segue:

- il confronto tra le simulazioni 1 e 3, oppure 2 e 4, indica la criticità della scelta del grado dell'approssimazione ad elementi finiti: infatti, a meno delle prime iterazioni in cui si è lontani dalla forma ottima, quando  $\boldsymbol{\mu}$  è tale che  $\Omega_o(\boldsymbol{\mu})$  è in un intorno di  $\hat{\Omega}$ , il funzionale costo ed il suo gradiente sono dominati dall'errore dell'approssimazione ad elementi finiti. Per questo motivo, la direzione di ricerca potrebbe non risultare più di discesa: ciò è particolarmente evidente nella simulazione 2, che viene interrotta dopo 15 iterazioni del ciclo di *backtracking* perché non risulta possibile verificare il criterio di Armijo. Analogamente, nella simulazione 1 si riscontra una prematura convergenza, a causa di decrementi eccessivamente ridotti del funzionale costo – sebbene la norma del gradiente non sia giunta alla tolleranza prescritta dal criterio di arresto – perché la direzione fornita dal gradiente è di discesa solo per un passo molto piccolo;
- il confronto tra la simulazione 3 e la simulazione 4 indica che nel caso del problema vincolato si ottiene una velocità di convergenza più rapida rispetto al caso non vincolato. Ciò, nei limiti dei risultati piuttosto scadenti, sembra essere confermato anche dal confronto tra la simulazione 1 e la simulazione 2;
- il confronto tra la simulazione 4 e la simulazione 5 mostra l'importanza della scelta dei punti di controllo della mappa *Free Form Deformation*: la scelta piuttosto innaturale di abilitare i punti di controllo ai vertici del cubo circoscritto alla sfera  $\hat{\Omega}$  comporta, dopo poche iterazioni, valori eccessivamente piccoli della norma del gradiente. Questo comportamento può essere dovuto alla distanza presente tra i punti di controllo abilitati e la superficie del dominio: per valori di  $\boldsymbol{\mu}$  sufficientemente piccoli, l'interpolazione dovuta ai polinomi di Bernstein

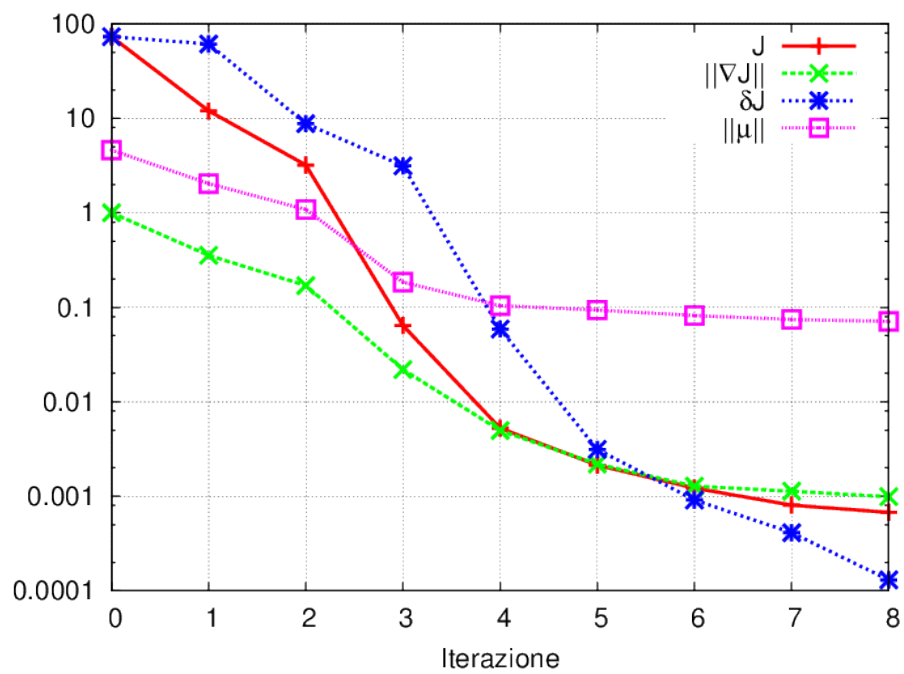
<sup>1</sup>Due CPU Dual Core AMD Opteron™ Processor 265, 1.8 GHz, 8 GB RAM. Server di calcolo offerto dal MOX, Dipartimento di Matematica, Politecnico di Milano.

<sup>2</sup>32 CPU Quad Core Intel® Xeon® Nehalem E5560, 2.4 GHz, 384 GB RAM. Cluster di calcolo offerto dal MOX, Dipartimento di Matematica, Politecnico di Milano.





(a) Convergenza della simulazione 3.



(b) Convergenza della simulazione 4.

Figura 5.5: Storia di convergenza delle simulazioni 3 e 4.

concentra le deformazioni imposte dalla mappa *Free Form Deformation* in un intorno dei punti di controllo, rendendole invece trascurabili sui nodi della mesh.

### 5.1.2 Caso test non stazionario

Si consideri ora il seguente problema non stazionario: dati  $f \in L^2(0, T; L^2(\mathbb{R}^n))$ ,  $g \in L^2(0, T; H^2(\mathbb{R}^n))$ , sia  $u = u(\Omega) \in L^2(0, T; H^1(\mathbb{R}^n)) \cap C^0(0, T; L^2(\mathbb{R}^n))$  soluzione del problema

$$\begin{cases} \partial_t u(\Omega) - \Delta u(\Omega) = f, & \text{in } \Omega \times (0, T), \\ u(\Omega) = g, & \text{su } \partial\Omega \times (0, T), \\ u(\Omega) = h, & \text{in } \Omega \times \{0\}. \end{cases}$$

Assegnato  $\hat{u} \in L^2(0, T; L^2(\mathbb{R}^n))$ , si consideri il seguente funzionale costo:

$$J(\Omega) = \int_0^T \int_{\Omega} |u(\Omega) - \hat{u}|^2;$$

il problema di ottimizzazione di forma è quindi

$$\hat{\Omega} = \arg \min_{\Omega \subset \mathbb{R}^n} J(\Omega). \quad (5.3)$$

Per brevità viene omessa l'analisi di esistenza e la scrittura delle condizioni di ottimalità: nel caso non stazionario questi argomenti sono comunque già stati affrontati, per il problema di stato governato dalle equazioni di Stokes, nella Sezione 3.3, e si possono ripetere analogamente anche per l'equazione del calore; in particolare, si può ricavare il seguente sistema di condizioni di ottimalità:

$$\begin{cases} \partial_t u(\Omega) - \Delta u(\Omega) = f, & \text{in } \Omega \times (0, T), \\ u(\Omega) = g, & \text{su } \partial\Omega \times (0, T), \\ u(\Omega) = h, & \text{in } \Omega \times \{0\}, \end{cases} \quad \text{problema di stato,}$$

$$\begin{cases} -\partial_t p(\Omega) - \Delta p(\Omega) = -2(u(\Omega) - \hat{u}), & \text{in } \Omega \times (0, T), \\ p(\Omega) = 0, & \text{su } \partial\Omega \times (0, T), \\ p(\Omega) = 0, & \text{in } \Omega \times \{T\}, \end{cases} \quad \text{problema aggiunto,}$$

$$dJ(\Omega; \boldsymbol{\theta}) = \int_0^T \int_{\partial\Omega} \left\{ |u(\Omega) - \hat{u}|^2 - \frac{\partial}{\partial \mathbf{n}} p(\Omega) \frac{\partial}{\partial \mathbf{n}} (u(\Omega) - g) \right\} \boldsymbol{\theta} \cdot \mathbf{n}, \quad \text{equazione di sensitività.}$$

Analogamente a quanto visto nella Sezione 3.3.4, il problema aggiunto è *backward* in tempo e  $p(\Omega)(t)$  dipende da  $u(\Omega)(t)$ , tramite il termine forzante. Dal punto di vista della risoluzione numerica, questo pone serie difficoltà nella gestione della memoria: infatti per risolvere il problema aggiunto è necessario memorizzare i valori della soluzione di stato a ciascun passo temporale, con notevole dispendio di risorse di memoria se il problema è tridimensionale e la griglia spaziale e temporale è fine. Una strategia per limitare questa difficoltà è stata proposta in [Ber98]: un possibile sviluppo per la libreria presentata nella Sezione 4.4.4 e nell'Appendice A è implementare questa strategia di riduzione dell'utilizzo di memoria.

Nel seguito si pone  $n = 3$ ,  $T = 2$ ,

$$\begin{aligned} f(x_1, x_2, x_3, t) &= \pi(x_3^2 - x_1^2 + 8) \cos(\pi t), \\ g(x_1, x_2, x_3, t) &= (x_1^2 + 2x_2^2 + 3x_3^2) \sin(\pi t), \\ h(x_1, x_2, x_3, t) &= 0, \\ \hat{u}(x_1, x_2, x_3, t) &= (x_3^2 - x_1^2 + 8) \sin(\pi t), \end{aligned}$$

#	Ordine FE	$\Delta t$	Punti di controllo	Vincolo di volume
<i>ii</i>	1	0.01	sfera	sì
<i>iv</i>	2	0.01	sfera	sì

#	Numero di iterazioni	$J^{(n)}$	$ J^{(n)} - J^{(n-1)} $	$\ \nabla_{\boldsymbol{\mu}} J^{(n)}\ _2$	$\ \boldsymbol{\mu}^{(n)}\ _2$
<i>ii</i>	6	$5.049 \times 10^{-1}$	* $9.082 \times 10^{-5}$	$8.570 \times 10^{-2}$	$3.596 \times 10^{-1}$
<i>iv</i>	10	$2.291 \times 10^{-1}$	* $8.631 \times 10^{-5}$	$4.536 \times 10^{-2}$	$7.954 \times 10^{-2}$

Tabella 5.2: Risultati del caso test non stazionario. La simulazione viene interrotta quando  $|J^{(n)} - J^{(n-1)}| < 10^{-4}$  oppure  $\|\nabla_{\boldsymbol{\mu}} J^{(n)}\|_2 < 10^{-3}$ : il criterio di arresto che ha causato la terminazione della simulazione è indicato con un asterisco. La configurazione dei punti di controllo denominata con ‘sfera’ si riferisce alla Figura 5.1(a).

in modo tale che, analogamente alla Sezione 5.1.1, il problema (5.3) ammetta una forma ottima in  $\widehat{\Omega} = B_2(\mathbf{0})$ , per la quale il funzionale costo è pari a  $J(\widehat{\Omega}) = 0$ .

Per quanto riguarda i risultati numerici, sono state eseguite due simulazioni sul cluster *idra* con 8 processori, utilizzando una mesh di circa 90.000 elementi di volume. In entrambi i casi i punti di controllo sono stati disposti nella configurazione di Figura 5.1(a) e, per aumentare la velocità di convergenza, è stato imposto un vincolo di volume.

Le due simulazioni differiscono per la scelta dell’ordine di discretizzazione spaziale (elementi finiti  $\mathbb{P}^1$  oppure  $\mathbb{P}^2$ ), mentre per la discretizzazione temporale è stato usato il metodo di Eulero implicito.

I risultati sono sostanzialmente in accordo con le considerazioni riportate per il caso stazionario (cfr. simulazioni 2 e 4). Si può notare, tuttavia, che:

- la simulazione *ii* è giunta a (prematura) convergenza, al contrario della simulazione 2 del caso stazionario che era stata bloccata da un’eccezione nel ciclo di *backtracking*;
- il miglioramento da *ii* a *iv* è molto meno marcato rispetto a quello tra 2 e 4 del caso stazionario, soprattutto per quanto riguarda i valori del funzionale costo e del suo gradiente. Ciò può essere dovuto ad un passo temporale non sufficientemente piccolo ( $\Delta t = 0.01$ ) e ad un ordine temporale basso: un’ulteriore diminuzione del passo temporale ed un aumento dell’ordine dello schema di avanzamento temporale potrebbero dunque portare a risultati migliori.

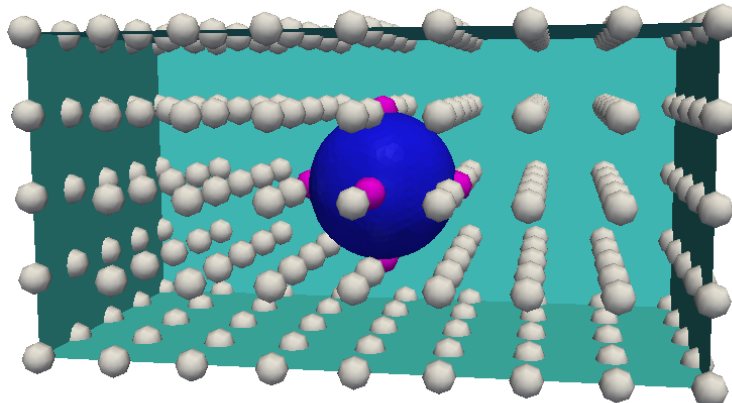


Figura 5.6: Dominio di riferimento e punti di controllo della mappa *Free Form Deformation*: il dominio di riferimento è dato dall'intersezione tra il complementare della palla (indicata in blu) ed il *bounding box*  $D$  (indicato in azzurro). I punti in grigio sono vincolati, quelli in rosa hanno un grado di libertà attivo nella direzione normale alla superficie della sfera.

## 5.2 Corpo immerso in un flusso di Stokes

In questa sezione si considera la minimizzazione dell'energia dissipata nel caso di un corpo immerso in un flusso di Stokes (geometrie di Figura 3.1(a)-(b)). Come anticipato nella Sezione 3.1, considerando una condizione di Dirichlet  $\mathbf{u} = \mathbf{u}_{in}$  anche sul bordo di outflow  $\Gamma_{out}$ , è possibile mostrare, nel caso stazionario, l'equivalenza con il problema di minimizzazione della resistenza (o *drag*) agente sul corpo immerso. Quest'ultimo problema è stato studiato da Pironneau negli anni '70 [Pir73]: in particolare, si può ricavare che, in un flusso di Stokes, il corpo tridimensionale che, a volume fissato, minimizza il drag è tale che la sua vorticità sulla superficie è costante in modulo. Supponendo inoltre che il problema abbia una *unica* soluzione, è possibile ottenere un risultato di simmetria più forte del Teorema 3.4: in questo caso il dominio ottimo risulta assialsimmetrico, con asse di simmetria nella direzione del flusso, ad estremità coniche e simmetrico rispetto all'asse del segmento che congiunge tali estremità (si veda la Figura 5.7); inoltre, all'ottimo,  $\alpha = \beta = 120^\circ$ . Questo risultato è stato confermato anche dalle simulazioni numeriche di Bourot in [Bou74], il quale ha trovato che il rapporto tra il *drag* della forma ottima e il *drag* della sfera di pari volume è 0.95425. L'obiettivo di questa sezione è confrontare i risultati che si ottengono con la libreria implementata a quelli di Pironneau [Pir73], Bourot [Bou74] e Ogawa [OK03].

I gradi di libertà e la geometria del dominio di riferimento della mappa *Free Form Deformation* sono indicati nella Figura 5.6. Si è scelto di utilizzare la geometria di Figura 3.1(a) (frontiera  $\Gamma_w$  a forma di parallelepipedo) e non quella di Figura 3.1(b) (frontiera  $\Gamma_w$  a forma di cilindro) perché, nel primo caso, è facile fissare la frontiera  $\Gamma_w$ : basta infatti scegliere come parallelepipedo  $D$  (*bounding box* della *Free Form Deformation*) il parallelepipedo individuato da  $\Gamma_{in}, \Gamma_w, \Gamma_{out}$  e non abilitare alcuno spostamento sulla frontiera<sup>3</sup>. Infine, occorre sottolineare che non ci si aspetta di convergere alla forma ottima individuata da Pironneau, operando deformazione mediante *FFD*: infatti, dato che la *FFD* opera deformazioni regolari del dominio di riferimento sferico, non sarà possibile ottenere le estremità coniche mostrate in Figura 5.7. Ci si aspetta comunque di ottenere

<sup>3</sup>Sfruttando la scrittura della mappa *Free Form Deformation* come perturbazione dell'identità (2.17) e la definizione dei polinomi di Bernstein, è possibile constatare che la frontiera di  $D$  può essere deformata solo da punti di controllo posti su  $\partial D$ .

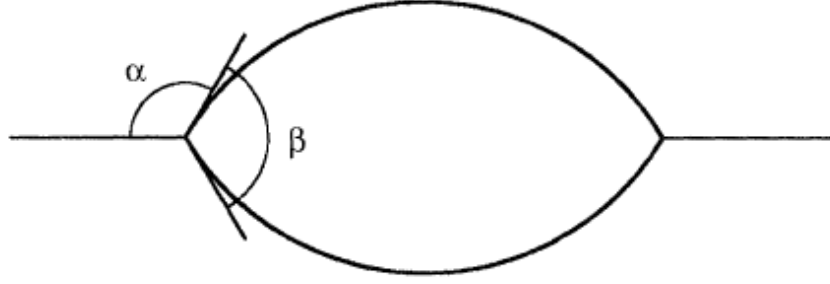


Figura 5.7: Sezione del profilo ottimo individuato da Pironneau in [Pir73], dove  $\alpha = \beta = 120^\circ$ . Tratto da [Ric95].

una forma sufficientemente affusolata e simile, ad una distanza opportuna dagli estremi, a quella mostrata in Figura 5.7.

### 5.2.1 Problema di Stokes stazionario

In questa sezione si considera dunque il seguente problema di Stokes stazionario: dato un profilo parabolico di inflow  $\mathbf{u}_{in} \in [H^{1/2}(\Gamma_{in})]^3$ , con velocità massima unitaria, le variabili di stato  $\mathbf{u} \in V_D(\Omega)$ ,  $p \in Q(\Omega)$  sono soluzione del seguente sistema:

$$\begin{cases} -\nu \Delta \mathbf{u}(\Omega) + \nabla p(\Omega) = \mathbf{0}, & \text{in } \Omega, \\ \operatorname{div} \mathbf{u}(\Omega) = 0, & \text{in } \Omega, \\ \mathbf{u}(\Omega) = \mathbf{u}_{in}, & \text{su } \Gamma_{in} \cup \Gamma_{out}, \\ \mathbf{u}(\Omega) = \mathbf{0}, & \text{su } \Gamma_w \cup \Gamma_c. \end{cases} \quad (5.4)$$

Denominando  $S$  la palla indicata in blu in Figura 5.6 ed  $S^c$  l'intersezione tra il suo complementare ed il *bounding box*  $D$ , indicato in azzurro, si considera il seguente insieme di forme ammissibili

$$\mathcal{O}_{ad} = \{\Omega \subset \mathbb{R}^3 : \Gamma_{in} \cup \Gamma_w \cup \Gamma_{out} \text{ fissato}, V(\Omega) = V(S^c)\},$$

cioè, dal punto di vista fisico, si considerano tutte le deformazioni del dominio di riferimento per le quali il volume del corpo immerso è costante. Si intende minimizzare l'energia dissipata

$$J(\Omega) = \frac{\nu}{2} \int_{\Omega} |\nabla \mathbf{u}(\Omega)|^2 d\mathbf{x},$$

che, grazie alle condizioni di Dirichlet anche su  $\Gamma_{out}$ , può essere espressa equivalentemente come il *drag* agente sul corpo immerso.

L'analisi del problema di ottimizzazione di forma

$$\widehat{\Omega} = \arg \min_{\Omega \in \mathcal{O}_{ad}} J(\Omega)$$

è già stata affrontata nella Sezione 3.2; in particolare le condizioni di ottimalità possono essere semplificate, come descritto nella Sezione 3.2.6, senza la necessità di introdurre alcun problema

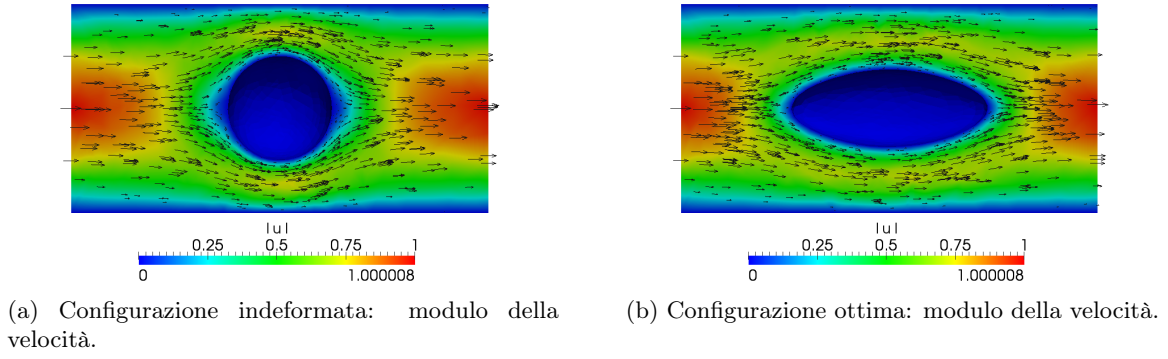


Figura 5.8: Velocità nella sezione longitudinale.

aggiunto:

$$\begin{cases} -\nu \Delta \mathbf{u}(\Omega) + \nabla p(\Omega) = \mathbf{0}, & \text{in } \Omega, \\ \operatorname{div} \mathbf{u}(\Omega) = 0, & \text{in } \Omega, \\ \mathbf{u}(\Omega) = \mathbf{u}_{in}, & \text{su } \Gamma_{in} \cup \Gamma_{out}, \\ \mathbf{u}(\Omega) = \mathbf{0}, & \text{su } \Gamma_w \cup \Gamma_c, \end{cases} \quad \text{problema di stato,}$$

$$dJ(\Omega; \boldsymbol{\theta}) = \int_{\Gamma_c} \left[ -\frac{\nu}{2} |\nabla \mathbf{u}(\Omega)|^2 \right] \boldsymbol{\theta} \cdot \mathbf{n}, \quad \text{equazione di sensitività.}$$

dove  $\boldsymbol{\theta}$  è tale che  $\boldsymbol{\theta} = \mathbf{0}$  su  $\Gamma_{in} \cup \Gamma_w \cup \Gamma_{out}$ . Inoltre, non è presente alcun moltiplicatore di Lagrange sul vincolo di volume perché si suppone che qualsiasi dominio deformato rispetti  $V([\mathbf{I} + \boldsymbol{\theta}](S^c)) = V(S^c)$ : infatti, se  $\boldsymbol{\theta}$  è lo spostamento associato alla mappa *FFD*, è possibile imporre anche questo secondo vincolo mediante una proiezione con l'algoritmo di Newton-Raphson introdotto nella Sezione 4.4.3.

Si vuole quindi approssimare la forma mediante parametrizzazione *FFD* e ricercare i parametri  $\boldsymbol{\mu}$  ottimi con il metodo di discesa del gradiente con proiezione, utilizzando la libreria implementata. Le simulazioni sono state eseguite sul cluster *idra*, con 8 processori, considerando la coppia di elementi finiti  $\mathbb{P}^1 - \mathbb{P}^1$ , con stabilizzazione, su una mesh di circa 60.000 elementi di volume.

Se  $\bar{\alpha}$ , il passo di discesa di primo tentativo del metodo del gradiente con proiezione, è sufficientemente grande si ottiene già alla prima iterazione una forma molto prossima a quella ottima: questo risultato è una importante conferma della bontà della *Free Form Deformation*, dato che la stessa rapidità di convergenza è stata riscontrata da Pironneau in [Pir73, KPT<sup>+</sup>00] usando una tecnica di *local boundary variation*. L'algoritmo giunge a convergenza alla quarta iterazione; inoltre la forma ottima e la convergenza non dipendono dalla scelta di  $\bar{\alpha}$ : anche dividendo il precedente  $\bar{\alpha}$  per 10 si ottiene convergenza comunque alla stessa forma ottima, in sei iterazioni.

Il confronto tra i profili di velocità nella sezione longitudinale è mostrato in Figura 5.8; alcune *streamlines* per il dominio di riferimento e la forma ottima sono mostrate in Figura 5.9. Come previsto da Pironneau, si ottiene una forma ottima assialsimmetrica, con asse di simmetria nella direzione del flusso: è importante sottolineare che questa simmetria non è stata imposta a priori, ma ci si è limitati ad imporre (oltre ai vincoli presenti in  $\mathcal{O}_{ad}$ ) un vincolo più debole, cioè che il centro della palla  $S$  rimanga fissato nella propria posizione per qualsiasi configurazione deformata.

La Figura 5.10 mostra il confronto con i risultati ottenuti da Pironneau [Pir73] e da Ogawa [OK03]: come previsto, non si riescono ad ottenere le estremità coniche mostrate in Figura 5.7; la forma appare leggermente più elongata rispetto a quella teorica. Il rapporto  $J(\hat{\Omega})/J(S)$  è pari a 0.929658417, migliore di quello trovato da Bourot in [Bou74]. Si ritiene che queste differenze possano essere causate da:

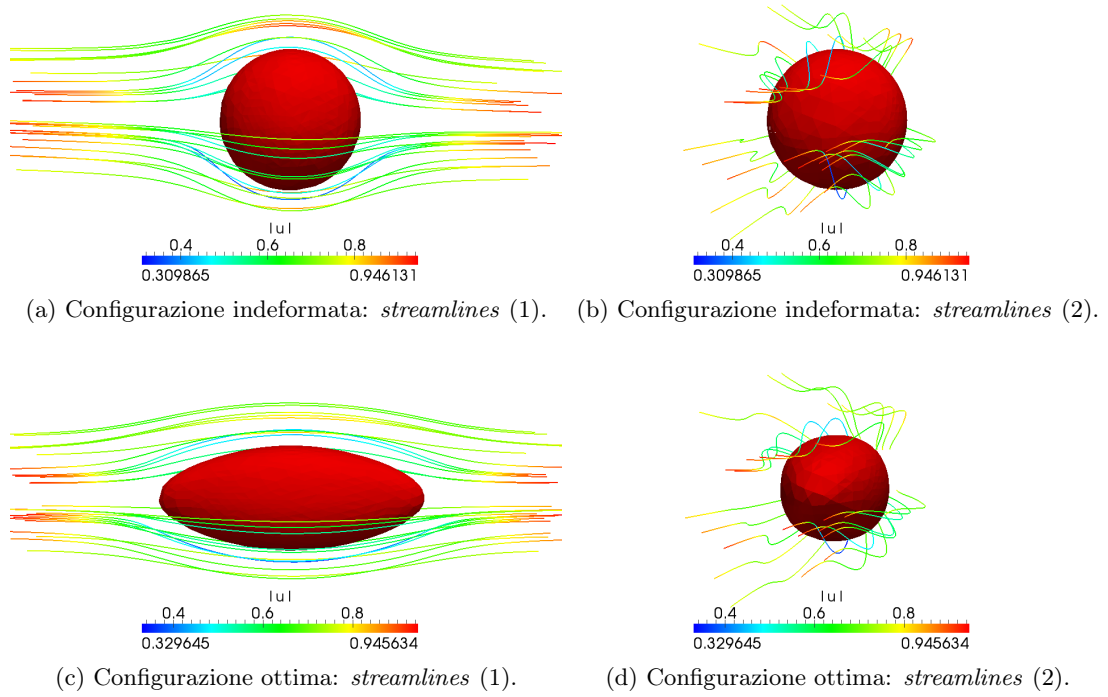
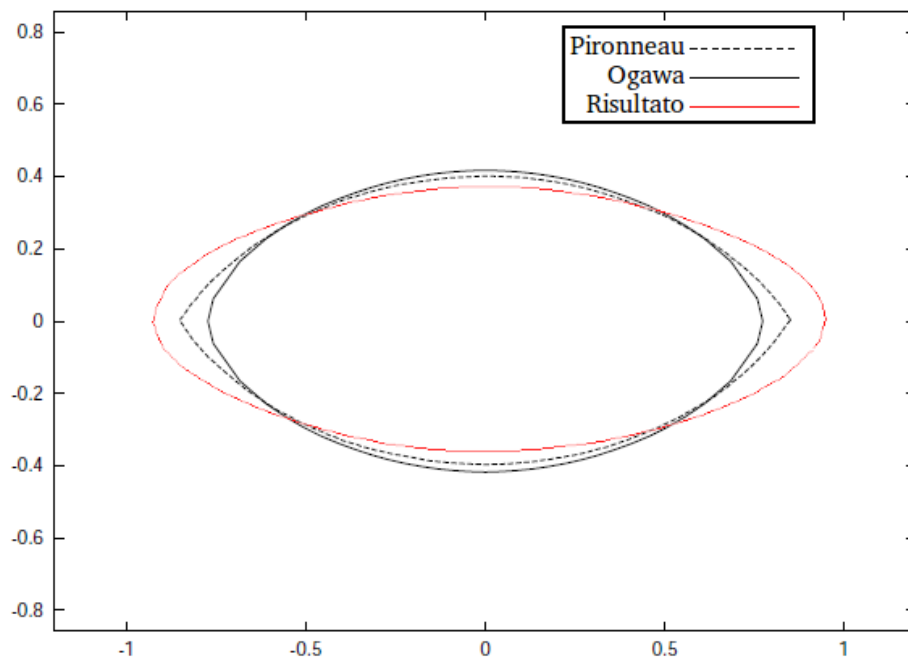
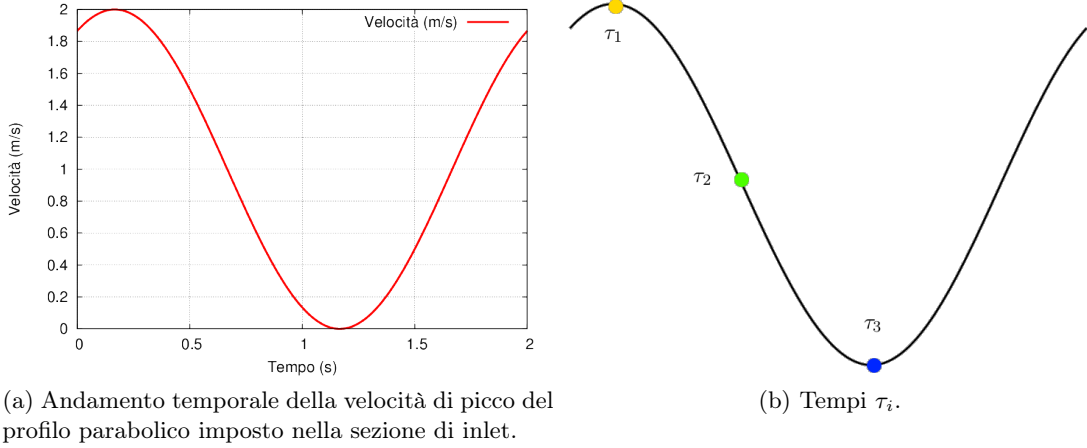
Figura 5.9: *Streamlines*.

Figura 5.10: Confronto con i risultati di Pironneau [Pir73] e Ogawa [OK03].



(a) Andamento temporale della velocità di picco del profilo parabolico imposto nella sezione di inlet.

(b) Tempi  $\tau_i$ .

Figura 5.11: Velocità di picco per la sezione di inlet.

- o scelta di elementi finiti di basso grado, in particolare per la velocità, che causano la valutazione del funzionale costo integrando quantità la cui approssimazione è piuttosto scadente ( $\nabla \mathbf{u}$  è solo costante a tratti);
- o pareti  $\Gamma_w$  e frontiere  $\Gamma_{in}$  e  $\Gamma_{out}$  troppo vicine al corpo immerso  $\Gamma_c$ : ciò è particolarmente evidente nella configurazione deformata (Figura 5.8(b)). Ogawa [OK03], ad esempio, considera un cubo  $D$  di lato pari a 10 volte il diametro di  $S$ , mentre in Figura 5.6 il rapporto tra il diametro di  $S$  e i lati del parallelepipedo  $D$  è solamente pari a 4 nella direzione parallela al flusso e 2 nelle altre direzioni. Non è stato tuttavia possibile effettuare simulazioni con *mesh* modificate secondo le dimensioni di Ogawa a causa dell'eccessiva richiesta di memoria durante risoluzione del problema di stato, dovuta ad un numero troppo elevato di elementi di volume.

## 5.2.2 Problema di Stokes non stazionario

In questa sezione si considera il seguente problema di Stokes non stazionario: noto il profilo parabolico di inflow  $\mathbf{u}_{in}(t) \in L^2(0, T; [H^{1/2}(\Gamma_{in})]^3)$ , il cui andamento temporale è mostrato in Figura 5.11(a), le variabili di stato  $\mathbf{u}(t) \in L^2(0, T; V_D(\Omega)) \cap C^0([0, T]; H(\Omega))$  e  $p(t) \in L^2(0, T; Q(\Omega))$  sono soluzione del seguente sistema:

$$\begin{cases} \partial_t \mathbf{u}(\Omega) - \nu \Delta \mathbf{u}(\Omega) + \nabla p(\Omega) = \mathbf{0}, & \text{in } \Omega \times (0, T), \\ \operatorname{div} \mathbf{u}(\Omega) = 0, & \text{in } \Omega \times (0, T), \\ \mathbf{u}(\Omega) = \mathbf{u}_{in}(t), & \text{su } \{\Gamma_{in} \cup \Gamma_{out}\} \times (0, T), \\ \mathbf{u}(\Omega) = \mathbf{0}, & \text{su } \{\Gamma_w \cup \Gamma_c\} \times (0, T), \\ \mathbf{u}(\Omega) = \mathbf{u}_0(\Omega), & \text{in } \Omega \times \{0\}, \end{cases}$$

dove  $\mathbf{u}_0(\Omega)$  è soluzione del problema di Stokes stazionario (5.4) con velocità di inlet data da  $\mathbf{u}_{in}(0)$ . Considerando il seguente insieme di forme ammissibili

$$\mathcal{O}_{ad} = \{\Omega \subset \mathbb{R}^3 : \Gamma_{in} \cup \Gamma_w \cup \Gamma_{out} \text{ fissato, } V(\Omega) = V(S^c)\},$$

ed il funzionale costo dato dall'energia dissipata

$$J(\Omega) = \frac{\nu}{2} \int_0^T \int_{\Omega} |\nabla \mathbf{u}(\Omega)|^2 \, dx \, dt,$$



si vuole risolvere il seguente problema di ottimizzazione di forma

$$\widehat{\Omega} = \arg \min_{\Omega \in \mathcal{O}_{ad}} J(\Omega),$$

la cui analisi è già stata affrontata nella Sezione 3.3; in particolare le condizioni di ottimalità, ricavate nella Sezione 3.3.4, sono date da:

$$\left\{ \begin{array}{l} \left\{ \begin{array}{ll} -\nu \Delta \mathbf{u}_0(\Omega) + \nabla p_0(\Omega) = \mathbf{0}, & \text{in } \Omega, \\ \operatorname{div} \mathbf{u}_0(\Omega) = 0, & \text{in } \Omega, \\ \mathbf{u}_0(\Omega) = \mathbf{u}_{in}(0), & \text{su } \Gamma_{in} \cup \Gamma_{out}, \\ \mathbf{u}_0(\Omega) = \mathbf{0}, & \text{su } \Gamma_w \cup \Gamma_c, \end{array} \right. & \text{problemi di stato,} \\ \left\{ \begin{array}{ll} \partial_t \mathbf{u}(\Omega) - \nu \Delta \mathbf{u}(\Omega) + \nabla p(\Omega) = \mathbf{0}, & \text{in } \Omega \times (0, T), \\ \operatorname{div} \mathbf{u}(\Omega) = 0, & \text{in } \Omega \times (0, T), \\ \mathbf{u}(\Omega) = \mathbf{u}_{in}(t), & \text{su } \{\Gamma_{in} \cup \Gamma_{out}\} \times (0, T), \\ \mathbf{u}(\Omega) = \mathbf{0}, & \text{su } \{\Gamma_w \cup \Gamma_c\} \times (0, T), \\ \mathbf{u}(\Omega) = \mathbf{u}_0(\Omega), & \text{in } \Omega \times \{0\}, \end{array} \right. & \\ \left\{ \begin{array}{ll} -\partial_t \mathbf{z} - \nu \Delta \mathbf{z} + \nabla q = -\nu \Delta \mathbf{u}(\Omega), & \text{in } \Omega \times (0, T), \\ \operatorname{div} \mathbf{z} = 0, & \text{in } \Omega \times (0, T), \\ \mathbf{z} = \mathbf{0}, & \text{su } \partial\Omega \times (0, T), \\ \mathbf{z} = \mathbf{0}, & \text{in } \Omega \times \{T\}, \end{array} \right. & \text{problemi aggiunti,} \\ \left\{ \begin{array}{ll} -\nu \Delta \mathbf{z}_0 + \nabla q_0 = \mathbf{z}|_{t=0}, & \text{in } \Omega, \\ \operatorname{div} \mathbf{z}_0 = 0, & \text{in } \Omega, \\ \mathbf{z}_0 = \mathbf{0}, & \text{su } \partial\Omega, \end{array} \right. & \\ dJ(\Omega; \boldsymbol{\theta}) = \int_0^T \int_{\Gamma_c} \left( \nu \nabla \mathbf{z} : \nabla \mathbf{u}(\Omega) - \frac{\nu}{2} |\nabla \mathbf{u}(\Omega)|^2 \right) \boldsymbol{\theta} \cdot \mathbf{n} + & \text{equazione di sensitività.} \\ + \int_{\Gamma_c} \nu \nabla \mathbf{z}_0 : \nabla \mathbf{u}_0(\Omega) \boldsymbol{\theta} \cdot \mathbf{n}, & \end{array} \right.$$

dove  $\boldsymbol{\theta}$  è tale che sia  $\boldsymbol{\theta} = \mathbf{0}$  su  $\Gamma_{in} \cup \Gamma_w \cup \Gamma_{out}$  sia il vincolo di volume è rispettato.

Si vuole quindi approssimare la forma mediante parametrizzazione *FFD* e ricercare i parametri  $\boldsymbol{\mu}$  ottimi con il metodo di discesa del gradiente con proiezione, utilizzando la libreria implementata. Le simulazioni sono state eseguite sul cluster *idra*, con 8 processori, considerando la coppia di elementi finiti  $\mathbb{P}^1 - \mathbb{P}^1$ , con stabilizzazione, su una mesh di circa 60.000 elementi di volume, ed avanzamento in tempo con il metodo di Eulero implicito, intervallo  $(0, T)$  pari ad un periodo completo del profilo di inlet e passo temporale  $\Delta t$  pari ad  $1/100$  di tale periodo.

La Figura 5.13 mostra i profili di velocità, nella sezione longitudinale, rispettivamente per la configurazione indeformata e per la configurazione ottima, che è stata trovata dopo sei iterazioni del metodo del gradiente con proiezione. La Figura 5.14 mostra le *streamlines* per le due geometrie, a diversi istanti temporali  $\tau_i$ , che sono indicati in Figura 5.11(b) e corrispondono ai tempi di massima, media e minima velocità.

Il rapporto tra l'energia dissipata all'ottimo e l'energia dissipata per la palla di pari volume è  $J(\widehat{\Omega})/J(S) = 0.924042727$ . La validità fisica della forma ottima trovata è purtroppo, almeno parzialmente, compromessa dalla scelta delle condizioni al bordo di outflow<sup>4</sup>: infatti, come mostrato dalle sottofigure (c) ed (f), ciò comporta la modellazione del flusso nel dominio  $\Omega$  al tempo  $t = \tau_3$  come un sistema chiuso, dato che la velocità su  $\Gamma_{in} \cup \Gamma_w \cup \Gamma_{out}$  è nulla. Un più corretto significato fisico della frontiera di outflow può comunque essere recuperato imponendo le consuete condizioni al contorno di Neumann; tuttavia è possibile evidenziare come l'ottimizzazione della

<sup>4</sup>.. scelte come condizioni di Dirichlet per eseguire il confronto con il caso stazionario.

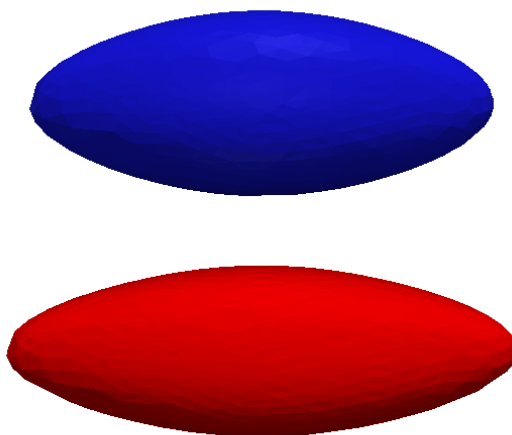


Figura 5.12: Confronto tra la forma ottima del caso stazionario (in blu) e quella del caso non stazionario (in rosso).

forma abbia contribuito a migliorare il flusso anche in questa condizione estrema: infatti, confrontando le sottofigure (b) ed (e), si può notare che già al tempo  $t = \tau_2$  nella forma ottima si ottiene una velocità massima (nelle zone a monte e a valle del corpo immerso) inferiore rispetto a quella rilevata nel dominio di riferimento. La diminuzione della velocità massima arriva anche al 30% circa al tempo  $t = \tau_3$  e ciò comporta, nell'istante in cui il sistema è chiuso, una notevole riduzione della ricircolazione.

Infine, la forma ottima del caso stazionario può essere confrontata con quella del caso non stazionario (Figura 5.12), e si può evidenziare che, nonostante il rapporto  $J(\hat{\Omega})/J(S)$  sia sostanzialmente uguale nei due casi, le due forme ottime non coincidono: la forma ottima del caso non stazionario risulta leggermente più elongata di quella del caso stazionario.

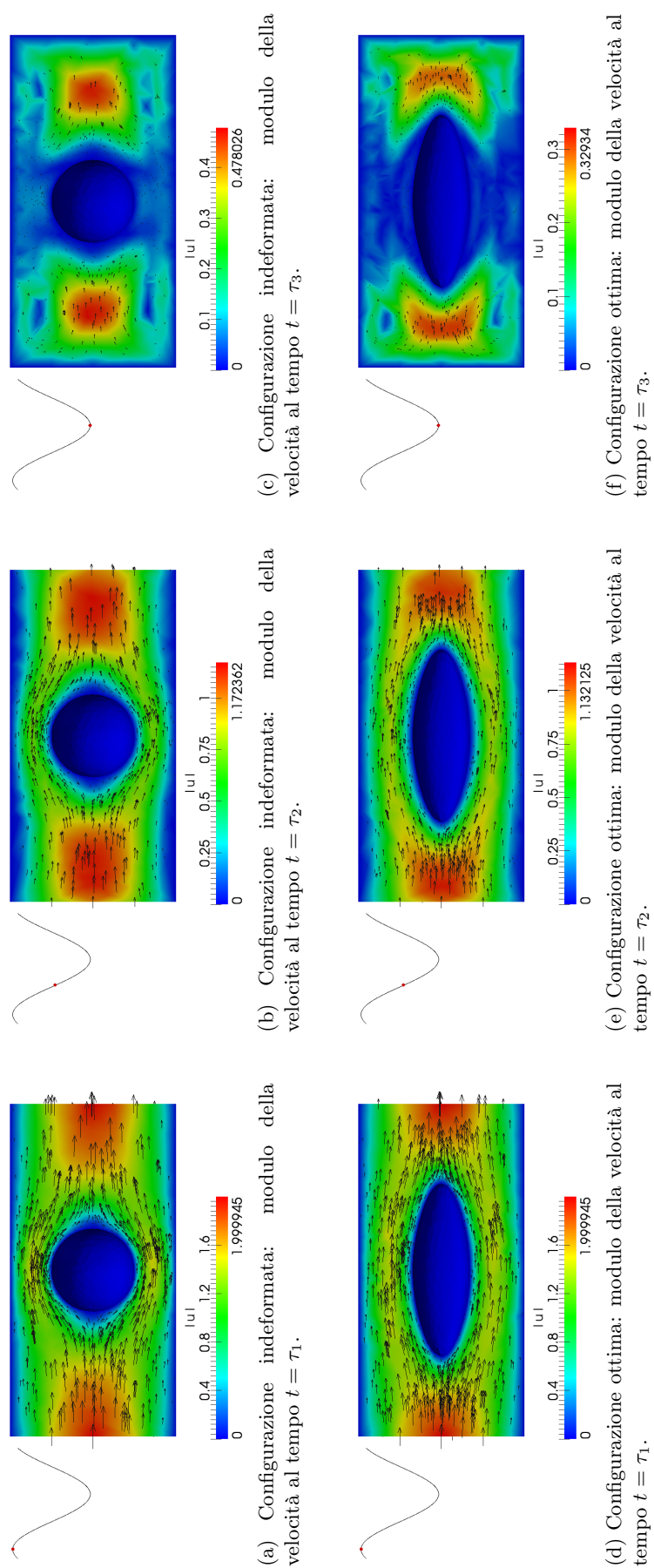


Figura 5.13: Velocità nella sezione longitudinale a diversi istanti temporali.

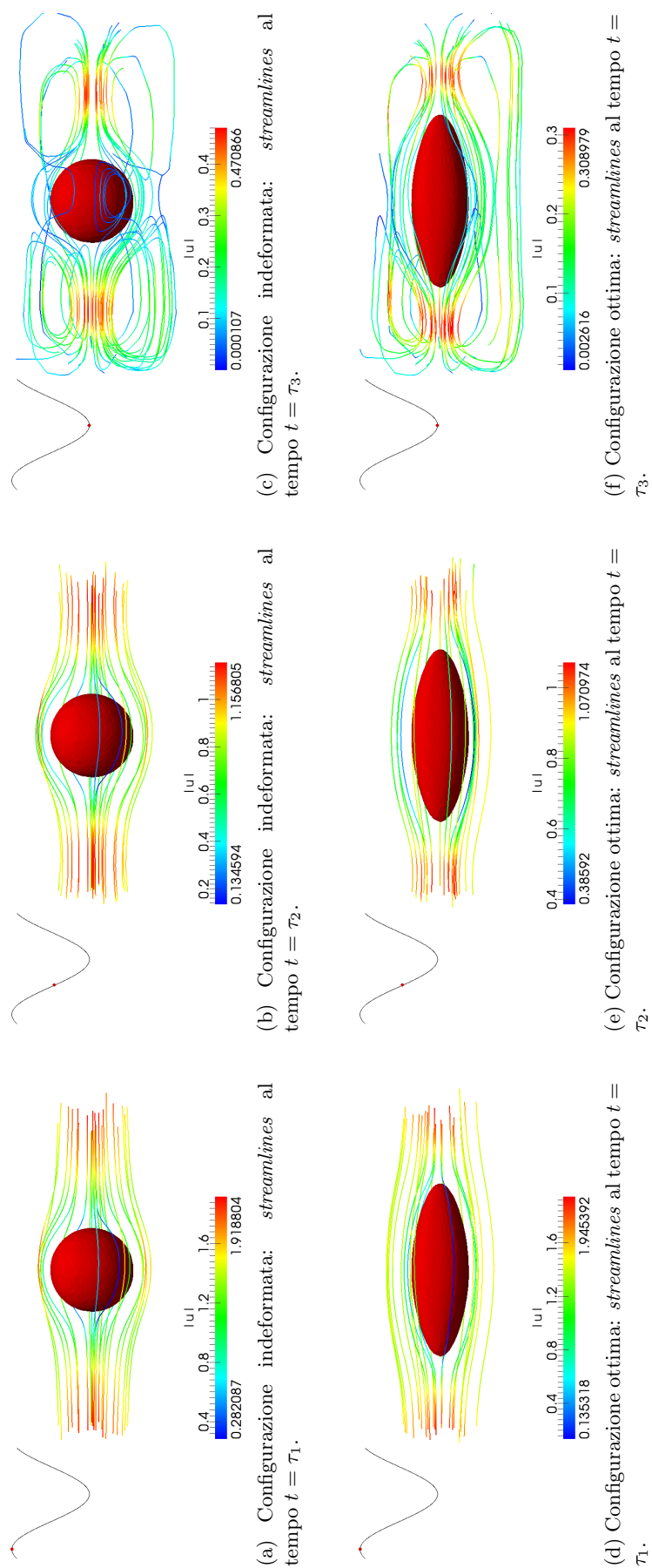


Figura 5.14: *Streamlines* a diversi istanti temporali.

### 5.3 Bypass aorto-coronarico

In questa sezione vengono presentati i risultati relativi al *design* ottimo della geometria di un bypass aorto-coronarico<sup>5</sup> di Figura 3.1(c); come già anticipato nella descrizione fornita nella Sezione 1.3, a questo problema sono stati dedicati numerosi studi negli ultimi decenni (ad esempio [DOYM11, WZJ<sup>+</sup>11, BDFD01, DBB01, PSC<sup>+</sup>08, BD00, LFB08, SCGT05]), che hanno messo in evidenza una forte influenza della geometria del condotto vascolare e dell’innesto sull’emodinamica e sul rischio di sviluppo di alcune patologie.

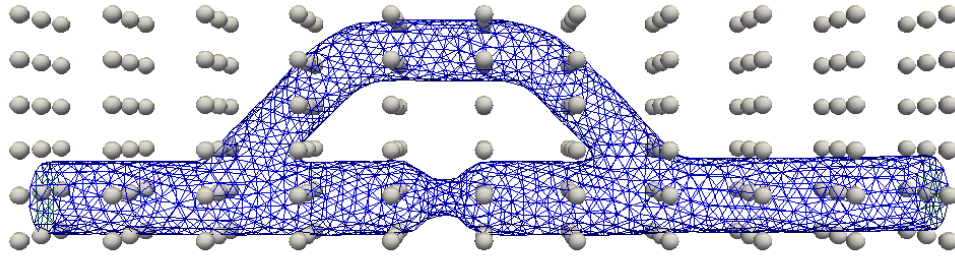
In particolare, il sangue è stato modellato come un fluido Newtoniano (per il quale si suppongono proprietà fisiche costanti: densità  $\rho = 1050 \text{ kg/m}^3$  e viscosità  $\mu = 3.5 \cdot 10^{-3} \text{ kg/(m} \cdot \text{s)}$ ). Dati tratti da [WZJ<sup>+</sup>11]); inoltre è possibile, in prima approssimazione, trascurare il termine convettivo e considerare dunque il modello di Stokes, grazie al ridotto diametro delle coronarie ed alle basse velocità. Le Sezioni 5.3.1 e 5.3.2 presentano i risultati per l’ottimizzazione di forma, rispetto al funzionale costo dell’energia dissipata, per il modello stazionario e, rispettivamente, non stazionario.

La risoluzione del problema di ottimizzazione di forma è stata effettuata mediante la libreria implementata, sfruttando una mappa *FFD* costruita sulla griglia di punti di controllo mostrata in Figura 5.15(a). La parametrizzazione mediante *Free Form Deformation* permette di ottimizzare su una vasta classe di configurazioni deformate, a differenza degli studi citati nella Sezione 1.3, che si basano sostanzialmente solo su variazioni degli angoli di innesto e della distanza del *graft* rispetto all’occlusione. Tuttavia, al fine di ottenere una procedura di ottimizzazione computazionalmente efficiente – in modo che il calcolo del gradiente del funzionale costo non sia eccessivamente oneroso – non è opportuno abilitare tutti i gradi di libertà indicati, ma è necessario selezionare solo i punti di controllo che si ritiene abbiano la maggiore influenza sul fenomeno fluidodinamico. Per una geometria complessa come quella del bypass aorto-coronarico questa scelta risulta meno immediata rispetto alla forma sferica delle Sezioni 5.1 e 5.2: per questo motivo sono state proposte diverse configurazioni di gradi di libertà (si vedano le Figure da 5.16 a 5.19 per il caso stazionario e le Figure 5.25 e 5.26 per il caso non stazionario). Un ulteriore sviluppo di questo lavoro potrebbe consistere nell’implementazione di tecniche automatiche per la selezione dei punti di controllo maggiormente significativi, ad esempio basate sulla sensitività, rispetto alla loro variazione, del funzionale costo da ottimizzare.

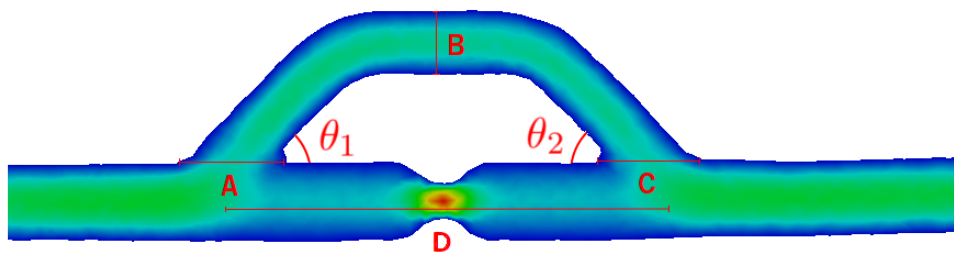
Una riduzione dimensionale è stata eseguita introducendo due diverse categorie di gradi di libertà: gradi di libertà di *ottimizzazione* e gradi di libertà di *deformazione*; i primi governano la dimensionalità del problema di ottimizzazione e dunque l’onere computazionale della simulazione, i secondi governano la dimensionalità della *FFD* e dunque la “capacità deformativa” della mappa. Infatti, è preferibile disporre di una simulazione con pochi gradi di libertà di ottimizzazione e molti gradi di libertà di deformazione: per questo motivo, gli spostamenti ottenuti sui punti di controllo con gradi di libertà di ottimizzazione (indicati con un cerchio nelle Figure da 5.16 a 5.18, 5.25 e 5.26) sono opportunamente propagati ai punti di controllo più prossimi (indicati con un quadrato del medesimo colore); il valore di questi spostamenti viene replicato anche sulle sezioni parallele a quella mostrata nelle Figure, e l’unione di tutti gli spostamenti imposti forma dunque l’insieme dei gradi di libertà di deformazione. La scelta dei gradi di libertà di ottimizzazione e la motivazione della propagazione delle informazioni verrà dettagliata caso per caso nelle prossime sezioni. Inoltre, supponendo di abilitare solamente spostamenti nelle direzioni degli assi del piano longitudinale<sup>6</sup>, la scelta di replicare i valori dei gradi di libertà sulle sezioni parallele al piano longitudinale – e dunque di imporre una simmetria rispetto ad esso – è giustificata, per lo meno nel caso stazionario, dall’esistenza di una forma ottima con tale simmetria, dimostrata nel Teorema 3.4.

<sup>5</sup>La mesh è stata fornita da Jean Bonnemain e Andrea Manzoni, MATHICSE - CMCS - EPFL.

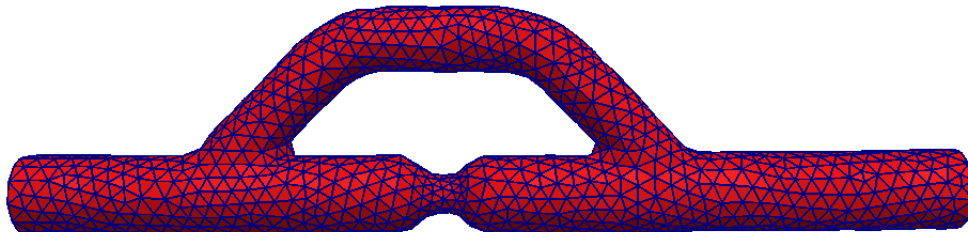
<sup>6</sup>Infatti appare ragionevole considerare deformazioni perpendicolari al piano longitudinale solo in presenza di una geometria più realistica, in cui si tenga conto della curvatura delle coronarie.



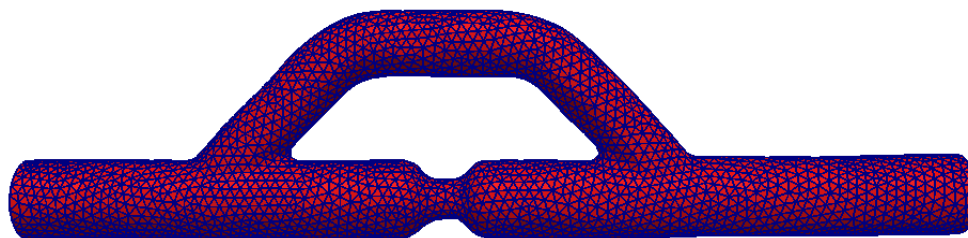
(a) Griglia dei punti di controllo, con  $K = 2$ ,  $L = 5$ ,  $M = 10$ , per un totale di 198 punti di controllo; considerando deformazioni solo nel piano longitudinale  $yz$  sono disponibili 396 gradi di libertà.



(b) Legenda distanze ed angoli.



(c) Mesh di circa 15.000 elementi di volume, 3.200 vertici e 1.400 vertici di bordo.



(d) Mesh di circa 70.000 elementi di volume, 13.000 vertici e 3.700 vertici di bordo.

Figura 5.15: Configurazione indeformata.

Infine, i risultati ottenuti con il modello lineare di Stokes verranno validati risolvendo un problema di Navier-Stokes non stazionario su ciascuna configurazione ottima (Sezione 5.3.3), discutendo anche come le geometrie trovate mediante l'ottimizzazione di forma possano comportare miglioramenti delle prestazioni mediche del *graft*, in particolare con una riduzione del processo di ristenosi.

### 5.3.1 Problema di Stokes stazionario

In questa sezione si considera un modello di Stokes stazionario per la descrizione dell'emodinamica nel bypass, in cui la velocità  $\mathbf{u} \in V_D(\Omega)$  e la pressione  $p \in Q(\Omega)$  del sangue sono soluzione del seguente sistema:

$$\begin{cases} -\nu \Delta \mathbf{u}(\Omega) + \nabla p(\Omega) = \mathbf{0}, & \text{in } \Omega, \\ \operatorname{div} \mathbf{u}(\Omega) = 0, & \text{in } \Omega, \\ \mathbf{u}(\Omega) = \mathbf{u}_{in}, & \text{su } \Gamma_{in}, \\ \mathbf{u}(\Omega) = \mathbf{0}, & \text{su } \Gamma_c, \\ \nu \frac{\partial \mathbf{u}(\Omega)}{\partial \mathbf{n}} - p(\Omega) \mathbf{n} = \mathbf{0}, & \text{su } \Gamma_{out}; \end{cases}$$

la condizione di inlet  $\mathbf{u}_{in} \in [H^{1/2}(\Gamma_{in})]^3$  è rappresentata da un profilo parabolico di modulo massimo  $u_{in}$ . Per geometrie in scala, in cui il raggio della sezione  $\Gamma_{in}$  è pari circa a 3 mm, un tipico valore è  $u_{in} = 0.144 \text{ m/s}$  (dati tratti da [WZJ+11, BD00])<sup>7</sup>.

Considerando il seguente insieme di forme ammissibili

$$\mathcal{O}_{ad} = \{\Omega \subset \mathbb{R}^3 : \Gamma_{in} \cup \Gamma_{out} \text{ fissato}\},$$

si intende minimizzare l'energia dissipata

$$J(\Omega) = \frac{\nu}{2} \int_{\Omega} |\nabla \mathbf{u}(\Omega)|^2 d\mathbf{x}.$$

L'analisi del problema di ottimizzazione di forma

$$\widehat{\Omega} = \arg \min_{\Omega \in \mathcal{O}_{ad}} J(\Omega)$$

è già stata affrontata nella Sezione 3.2; in particolare le condizioni di ottimalità possono essere semplificate senza la necessità di introdurre alcun problema aggiunto, come descritto nella Sezione 3.2.6, alla quale si rimanda per la scrittura dalle condizioni di ottimalità.

Si vuole quindi rappresentare le deformazioni mediante una parametrizzazione *FFD* e ricercare i parametri  $\boldsymbol{\mu}$  ottimi con il metodo di discesa del gradiente con proiezione, utilizzando la libreria implementata. Le simulazioni sono state eseguite sul cluster *idra*, con 8 processori, considerando la coppia di elementi finiti stabili  $\mathbb{P}^2 - \mathbb{P}^1$  su una mesh di circa 15.000 elementi di volume (Figura 5.15(c)). Questo permette di ottenere tempi computazionali sufficientemente ridotti: ciascuna risoluzione del problema di stato e successivo calcolo del funzionale costo richiede circa 5 minuti (si ricordi che, a ciascuna iterazione, può essere necessario risolvere più volte il problema di stato fino a quando il criterio di Armijo non sia verificato), ed il calcolo del gradiente del funzionale costo rispetto al singolo grado di libertà richiede circa 1 minuto (si ricordi anche che non è richiesta la risoluzione di alcun problema aggiunto, con un risparmio quindi di circa 5 minuti a ciascuna iterazione).

La Tabella 5.3 riassume i risultati ottenuti per le seguenti configurazioni di punti di controllo:

- (a) *Figura 5.16 – simulazione a*: si considerano 8 gradi di libertà di ottimizzazione, associati agli spostamenti in orizzontale ed in verticale dei quattro punti di controllo indicati in Figura 5.16

<sup>7</sup>Tutti i grafici riportati in questa sezione sono adimensionalizzati rispetto alla velocità  $u_{in}$ .

#	Gradi di libertà di ottimizzazione	Gradi di libertà di deformazione	Variazione funzionale costo	Numero di iterazioni
<i>a</i>	8	72	-10%	5
<i>b</i>	8	84	-12%	37
<i>c</i>	16	72	-23%	85
<i>d'</i>	12	60	-18%	35
<i>d''</i>	12	60	-26%	77

#	$\theta_1$	$\theta_2$	Variazione diametro A	Variazione diametro B	Variazione diametro C	Variazione distanza D
<i>a</i>	45°	60°	+20%	0%	0%	+4%
<i>b</i>	40°	40°	+40%	0%	+20%	+30%
<i>c</i>	50°	55°	+40%	0%*	+20%	+30%
<i>d'</i>	65°	55°	+30%	0%*	+10%	-10%
<i>d''</i>	55°	85°	+45%	0%*	+20%	-10%

Tabella 5.3: Risultati dell'ottimizzazione di forma di un bypass aorto-coronarico, con flusso stazionario. La legenda delle distanze e degli angoli è mostrata in Figura 5.15(b). La presenza di un asterisco indica che tale variazione di diametro è stata ottenuta mediante un vincolo.

con un cerchio. Il valore dello spostamento dei punti di controllo associati ai gradi di libertà di ottimizzazione viene assegnato anche ai punti di controllo indicati con un quadrato dello stesso colore e replicato sulle sezioni parallele al piano longitudinale, per un totale di 72 gradi di libertà di deformazione. La selezione di soli 8 gradi di libertà permette di ridurre drasticamente la dimensione del problema di ottimizzazione; inoltre, le variabili selezionate consentono di effettuare una variazione sia degli angoli di incidenza degli innesti sia della distanza compresa tra la stenosi e la linea di sutura. Queste variabili tuttavia non sono sufficienti a deformare il *graft* compatibilmente con le indicazioni cliniche, che prevedono di mantenerne costante il diametro (con possibili variazioni solo in prossimità degli innesti): infatti una deformazione verticale comporta sicuramente una variazione di diametro nel tratto orizzontale del *graft*<sup>8</sup>. Per questo motivo sono stati considerati come gradi di libertà di deformazione anche gli spostamenti agenti sui punti di controllo indicati con un quadrato; in particolare, il vincolo di diametro viene approssimato imponendo gli stessi spostamenti a punti di controllo immediatamente adiacenti in verticale.

(b) *Figura 5.17 – simulazione b*: si considerano 8 gradi di libertà di ottimizzazione, associati agli spostamenti in orizzontale ed in verticale dei quattro punti di controllo indicati in Figura 5.17 con un cerchio. Il valore dei gradi di libertà di ottimizzazione viene assegnato anche ai punti di controllo indicati con un quadrato dello stesso colore secondo la seguente regola:

1. se il punto di controllo indicato con un quadrato si trova al di sopra di quello indicato con un cerchio, si impone spostamento orizzontale nullo e spostamento verticale pari al grado di libertà di ottimizzazione verticale;
2. se il punto di controllo indicato con un quadrato si trova al di sotto di quello indicato con un cerchio, si impone spostamento verticale nullo e spostamento orizzontale pari al grado di libertà di ottimizzazione orizzontale.

<sup>8</sup>Infatti la faccia superiore del tratto orizzontale del bypass è tangente a  $\partial D$  e quindi deformabile solo da gradi di libertà posti su  $\partial D$  stesso. Non appare ragionevole considerare, per scopi computazionali, un parallelepipedo  $D'$  come quello mostrato in blu nella Figura 3.1(c), perché ciò limiterebbe l'entità delle deformazioni verticali, ma piuttosto è preferibile abilitare gradi di libertà anche su  $\partial D$ .



Gli spostamenti così ottenuti vengono replicati anche sulle sezioni parallele al piano longitudinale, per un totale di 84 gradi di libertà di deformazione. Come nel caso precedente, gli 8 gradi di libertà consentono di effettuare una variazione sia degli angoli di incidenza degli innesti sia della distanza compresa tra la stenosi e la linea di sutura. La scelta di abilitare punti di controllo collocati al di sopra di quelli indicati con un cerchio è giustificata dalle stesse considerazioni della simulazione *a*, notando tuttavia che, al fine di mantenere costante il diametro del *graft* nel tratto orizzontale, è sufficiente considerare solo i gradi di libertà verticali; la scelta di abilitare punti di controllo collocati al di sotto di quelli indicati con un cerchio è giustificata dalla possibilità di ottenere, in questo modo, una configurazione più regolare<sup>9</sup>.

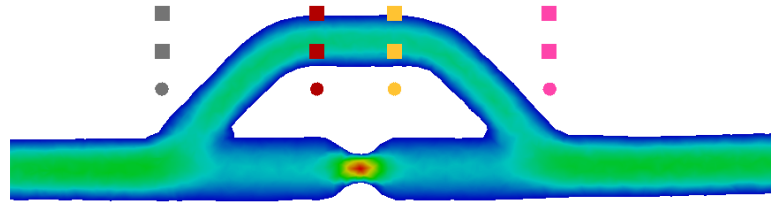
- (c) *Figura 5.18 – simulazione c*: in questo caso si considerano 16 gradi di libertà di ottimizzazione, associati agli spostamenti in orizzontale ed in verticale dei sei punti di controllo indicati in Figura 5.18 con un cerchio nero ed allo spostamento in orizzontale dei quattro punti di controllo indicati con un cerchio colorato. Il valore dei gradi di libertà di ottimizzazione, sia orizzontali sia verticali, viene assegnato anche ai punti di controllo indicati con un quadrato dello stesso colore. Gli spostamenti così ottenuti sono replicati sulle sezioni parallele al piano longitudinale, per un totale di 72 gradi di libertà di deformazione. Rispetto alle configurazioni delle simulazioni *a* e *b*, i punti di controllo indicati con un cerchio nero permettono anche di ottenere deformazioni più complesse, per le quali viene garantito che il diametro del tratto orizzontale del *graft* rimanga invariato imponendo un vincolo (lineare nelle componenti di  $\mu$ ) che mantenga costante la distanza *B* di Figura 5.15(b). La scelta di abilitare gli spostamenti orizzontali dei punti di controllo indicati con un quadrato è giustificata con le stesse considerazioni della simulazione *b*.
- (d) *Figura 5.19 – simulazioni d' e d''*: questa configurazione è stata ottenuta da quella della simulazione *c*, disabilitando rispetto all'originale gli spostamenti in orizzontale dei punti di controllo indicati con un cerchio nero in Figura 5.19, che causano un rigonfiamento innaturale nel *graft* ottimo della simulazione *c* (si veda la Figura 5.18(b)). Si considerano dunque solo 12 gradi di libertà di ottimizzazione, associati agli spostamenti in orizzontale ed in verticale dei due punti di controllo indicati con un asterisco nero, allo spostamento in verticale dei quattro punti di controllo indicati con un cerchio nero ed allo spostamento in orizzontale dei quattro punti di controllo indicati con un cerchio colorato. Il valore dei gradi di libertà di ottimizzazione, sia orizzontali sia verticali, viene assegnato anche ai punti di controllo indicati con un quadrato dello stesso colore. Gli spostamenti così ottenuti sono replicati sulle sezioni parallele al piano longitudinale, per un totale di 60 gradi di libertà di deformazione. Per questa configurazione vengono riportate le forme a due diverse iterazioni del ciclo di ottimizzazione perché, a differenza delle simulazioni precedenti, è stata riscontrata una anomalia durante la convergenza del processo di ottimizzazione: la norma del gradiente del funzionale costo, dopo una iniziale diminuzione, nelle ultime iterazioni tende ad aumentare. Per questo motivo viene riportata sia una configurazione nella fase di iniziale discesa (simulazione *d'*), sia la configurazione al termine del processo iterativo (simulazione *d''*).

Le Figura 5.20 mostra il modulo della velocità, nella sezione longitudinale, per la configurazione indeformata e per le configurazioni ottime delle tre simulazioni. Il flusso imposto dal profilo di inlet si divide, nella regione dell'innesto prossimale, tra il *graft* e l'arteria occlusa, come evidenziato dalle *streamlines* (Figura 5.22); in particolare, la velocità del sangue che percorre il *graft* è più elevata di quella del flusso che prosegue nella coronaria. La massima velocità è raggiunta nella zona della stenosi: in tutti i casi (*a*, *b*, *c*, *d'*, *d''*), l'ottimizzazione della forma porta a diminuire

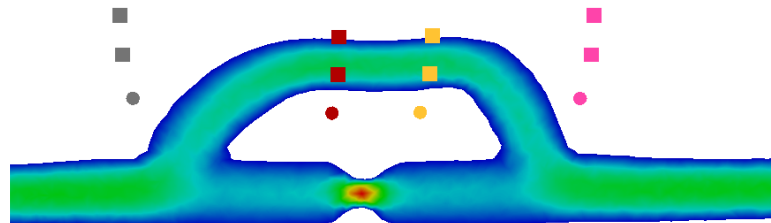
<sup>9</sup>Infatti la faccia inferiore della coronaria è tangente a  $\partial D$  e quindi deformabile solo da gradi di libertà posti su  $\partial D$  stesso. Mantenendo tale faccia fissata, la presenza di spostamenti orizzontali sulla faccia superiore della coronaria può contribuire alla perdita di regolarità della *mesh*, perché alcuni elementi di volume possono risultare particolarmente "schiacciati": per questo motivo è opportuno imporre una maggiore omogenità degli spostamenti imposti sulla superficie della coronaria.

la velocità massima rispetto alla configurazione indeformata, grazie ad un miglior flusso nel *graft*. Infine, come mostrato nella Figura 5.21, la velocità mantiene il profilo parabolico del dato di inlet, almeno a distanza sufficiente dagli innesti.

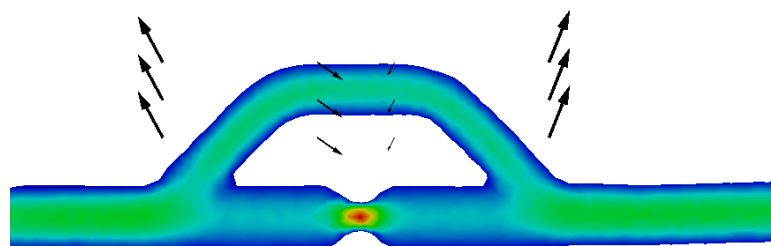
La Figura 5.23 mostra l'andamento dello sforzo di taglio a parete (WSS) nella regione a valle della stenosi; questa analisi è particolarmente rilevante per scopi clinici perché, come descritto nella Sezione 1.3, un basso valore dello sforzo di taglio a parete è un possibile indicatore dell'aterosclerosi. In tutte le figure si nota la presenza, nella coronaria, di una zona di basso WSS a monte dell'innesto: questa regione potrebbe dunque essere fortemente soggetta al processo aterosclerotico, che comporta un progressivo aumento delle dimensioni della stenosi fino alla completa occlusione dell'arteria. A valle dell'innesto è invece presente una zona ad elevato WSS. Tuttavia, a causa della linearità del modello fluidodinamico, il WSS si mantiene sostanzialmente costante in questa regione, con un comportamento notevolmente semplificato rispetto ai risultati citati nella Sezione 1.3: per questo motivo le forme ottime saranno validate nella Sezione 5.3.3 considerando un modello di Navier-Stokes. Infine, il confronto tra i valori dello sforzo di taglio a parete nel *graft* indica che è sono preferibili angoli di innesto piccoli (Figura 5.23(a) e (c)): infatti le geometrie delle sottofigure (b), (d) e (f) (ed in modo meno marcato, anche (e)), caratterizzate da un angolo  $\theta_2 > 45^\circ$  (per la definizione di  $\theta_2$  si veda la Figura 5.15(b)), contribuiscono alla formazione di regioni a basso WSS anche nella parte terminale del *graft*.



(a) Configurazione indeformata (sezione).

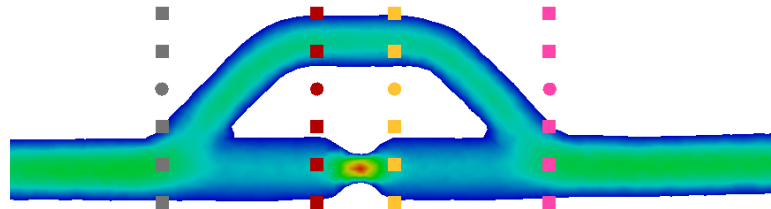


(b) Configurazione ottima (sezione).

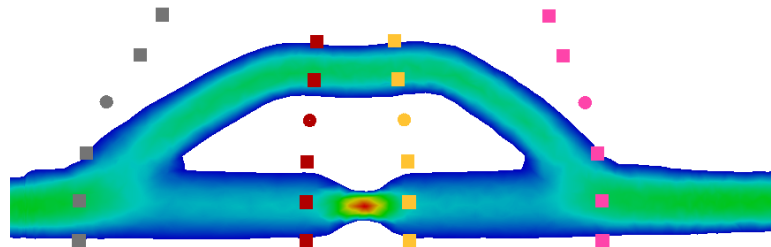


(c) Spostamento imposto sui punti di controllo abilitati per ottenere la configurazione ottima (sezione).

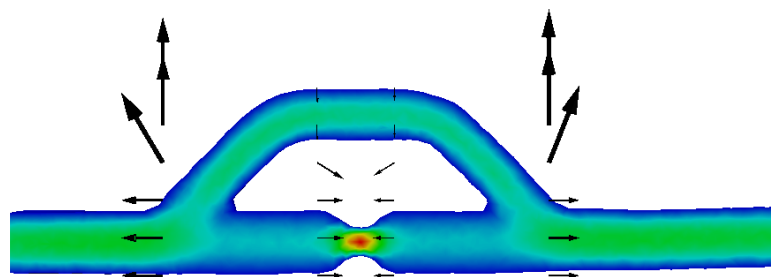
Figura 5.16: Simulazione *a*: rappresentazione della configurazione ottima, dei punti di controllo con gradi di libertà di ottimizzazione (indicati con un cerchio) e dei punti di controllo con gradi di libertà di deformazione (unione dei precedenti con i punti di controllo indicati con un quadrato, da replicarsi anche per le altre due sezioni parallele). I gradi di libertà di deformazione sono legati ai gradi di libertà di ottimizzazione del medesimo colore.



(a) Configurazione indeformata (sezione).

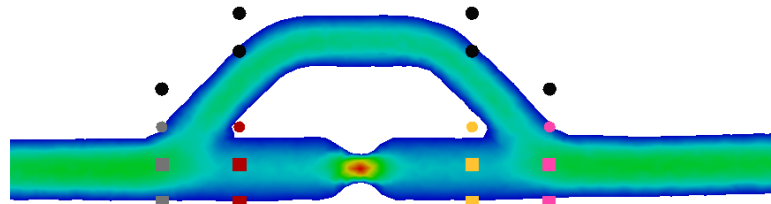


(b) Configurazione ottima (sezione).

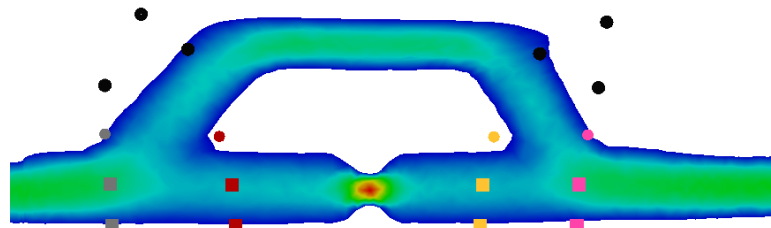


(c) Spostamento imposto sui punti di controllo abilitati per ottenere la configurazione ottima (sezione).

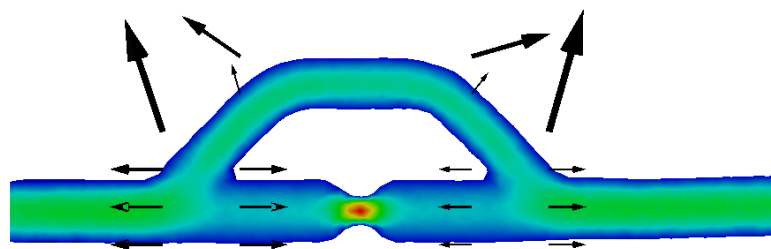
Figura 5.17: Simulazione *b*: rappresentazione della configurazione ottima, dei punti di controllo con gradi di libertà di ottimizzazione (indicati con un cerchio) e dei punti di controllo con gradi di libertà di deformazione (unione dei precedenti con i punti di controllo indicati con un quadrato, da replicarsi anche per le altre due sezioni parallele). I gradi di libertà di deformazione sono legati ai gradi di libertà di ottimizzazione del medesimo colore.



(a) Configurazione indeformata (sezione).

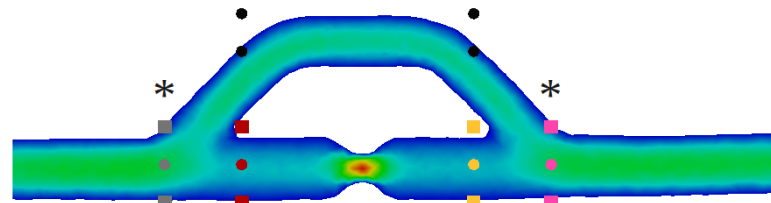


(b) Configurazione ottima (sezione).



(c) Spostamento imposto sui punti di controllo abilitati per ottenere la configurazione ottima (sezione).

Figura 5.18: Simulazione *c*: rappresentazione della configurazione ottima, dei punti di controllo con gradi di libertà di ottimizzazione (indicati con un cerchio) e dei punti di controllo con gradi di libertà di deformazione (unione dei precedenti con i punti di controllo indicati con un quadrato, da replicarsi anche per le altre due sezioni parallele). I gradi di libertà di deformazione sono legati ai gradi di libertà di ottimizzazione del medesimo colore.



(a) Configurazione indeformata (sezione).

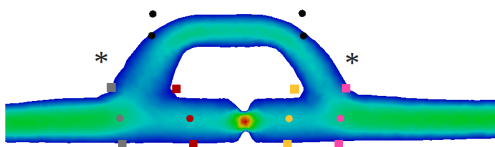
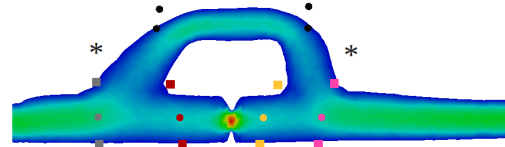
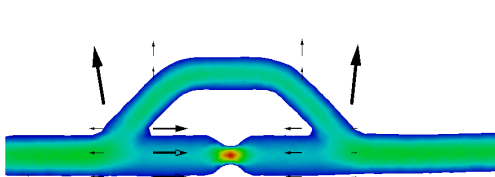
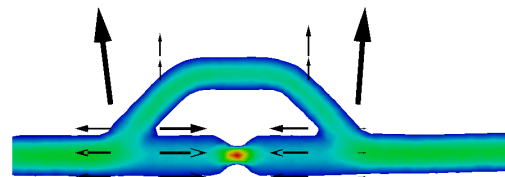
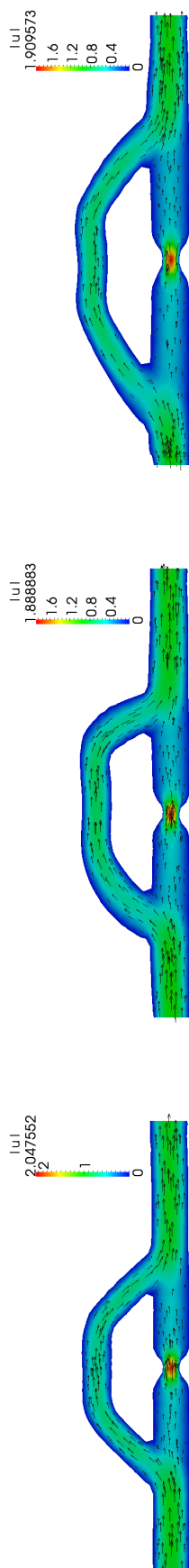
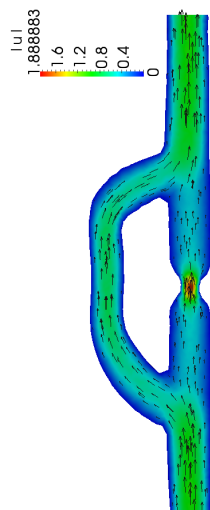
(b) Configurazione ottima, simulazione  $d'$  (sezione).(c) Configurazione ottima, simulazione  $d''$  (sezione).(d) Spostamento imposto sui punti di controllo abilitati per ottenere la configurazione ottima della simulazione  $d'$  (sezione).(e) Spostamento imposto sui punti di controllo abilitati per ottenere la configurazione ottima della simulazione  $d''$  (sezione).

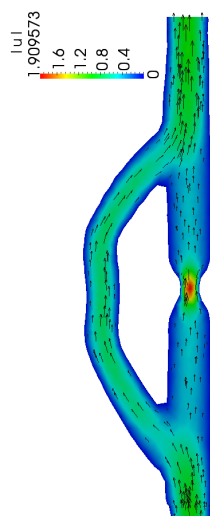
Figura 5.19: Simulazioni  $d'$  e  $d''$ : rappresentazione della configurazione ottima, dei punti di controllo con gradi di libertà di ottimizzazione (indicati con un cerchio o un asterisco) e dei punti di controllo con gradi di libertà di deformazione (unione dei precedenti con i punti di controllo indicati con un quadrato, da replicarsi anche per le altre due sezioni parallele). I gradi di libertà di deformazione sono legati ai gradi di libertà di ottimizzazione del medesimo colore.



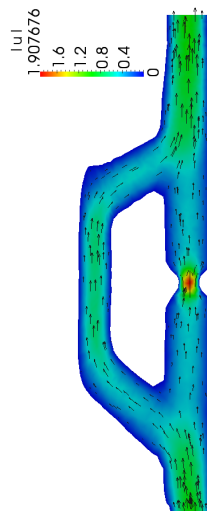
(a) Configurazione indeformata: modulo della velocità.



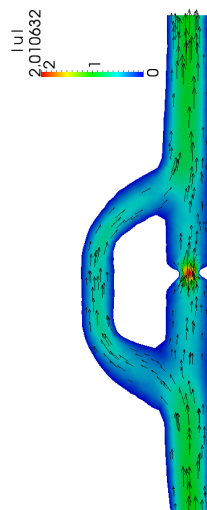
(b) Configurazione ottima, simulazione  $a$ : modulo della velocità.



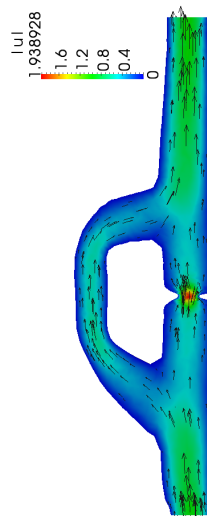
(c) Configurazione ottima, simulazione  $b$ : modulo della velocità.



(d) Configurazione ottima, simulazione  $c$ : modulo della velocità.



(e) Configurazione ottima, simulazione  $d'$ : modulo della velocità.



(f) Configurazione ottima, simulazione  $d''$ : modulo della velocità.

Figura 5.20: Velocità nella sezione longitudinale.

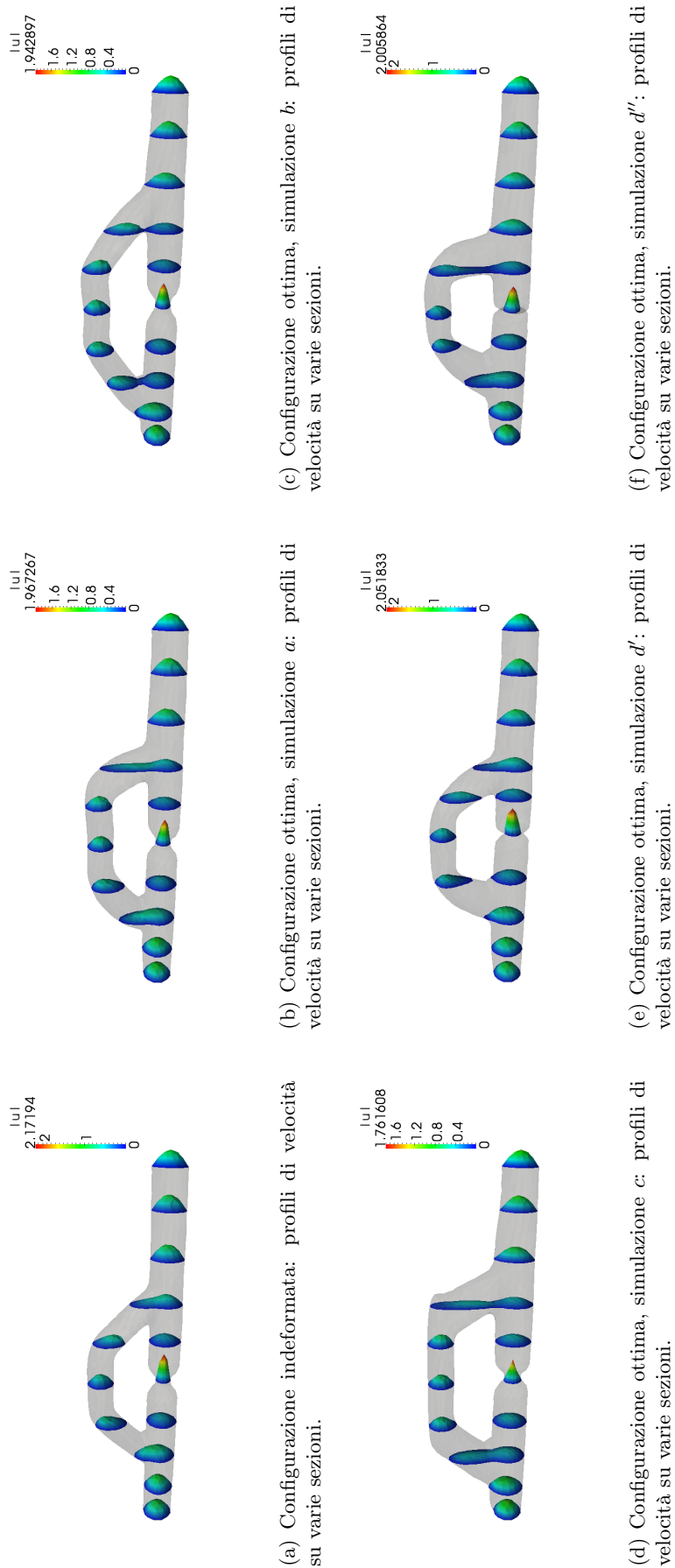
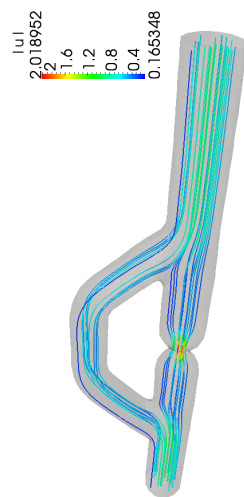
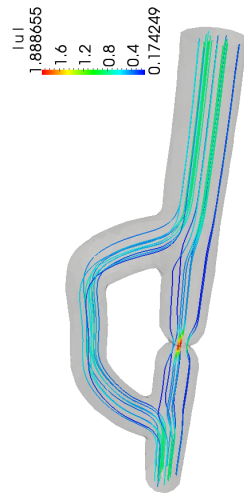
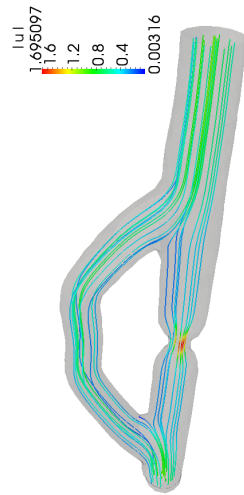
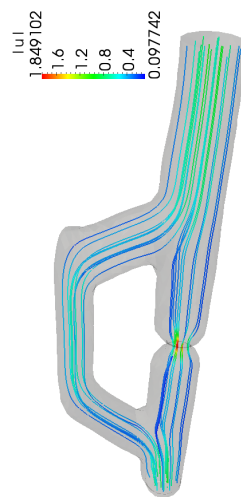
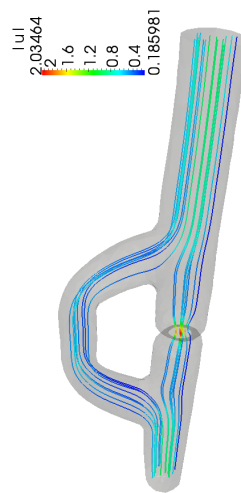
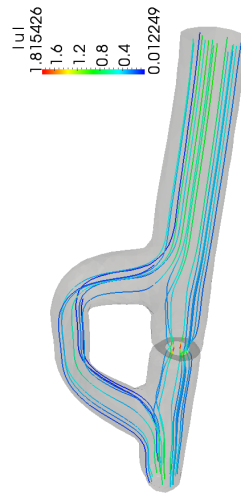


Figura 5.21: Profili di velocità in diverse sezioni.



(a) Configurazione indeformata: *streamlines*.(b) Configurazione ottima, simulazione *a*: *streamlines*.(c) Configurazione ottima, simulazione *b*: *streamlines*.(d) Configurazione ottima, simulazione *c*: *streamlines*.(e) Configurazione ottima, simulazione *d*: *streamlines*.(f) Configurazione ottima, simulazione *d''*: *streamlines*.Figura 5.22: *Streamlines*.

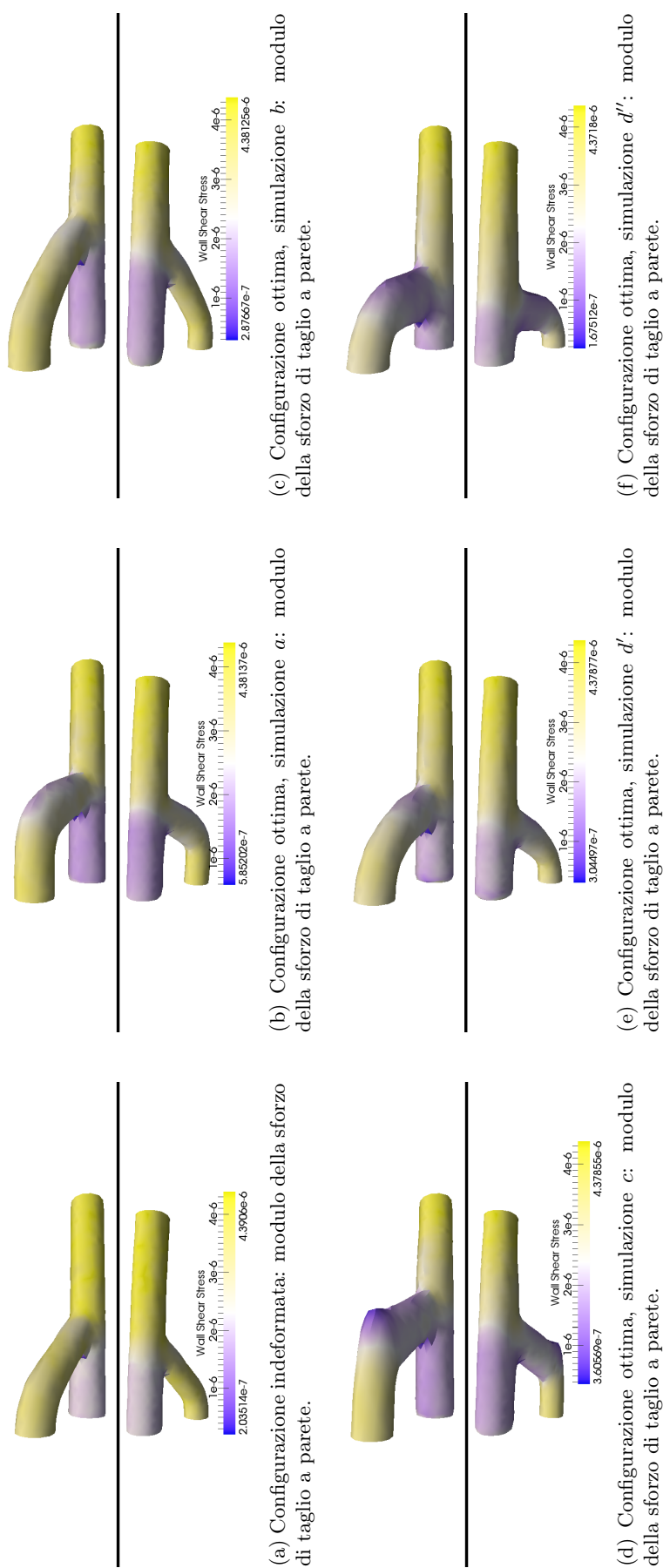


Figura 5.23: Modulo della sforzo di taglio a parete nella regione a valle della stenosi.

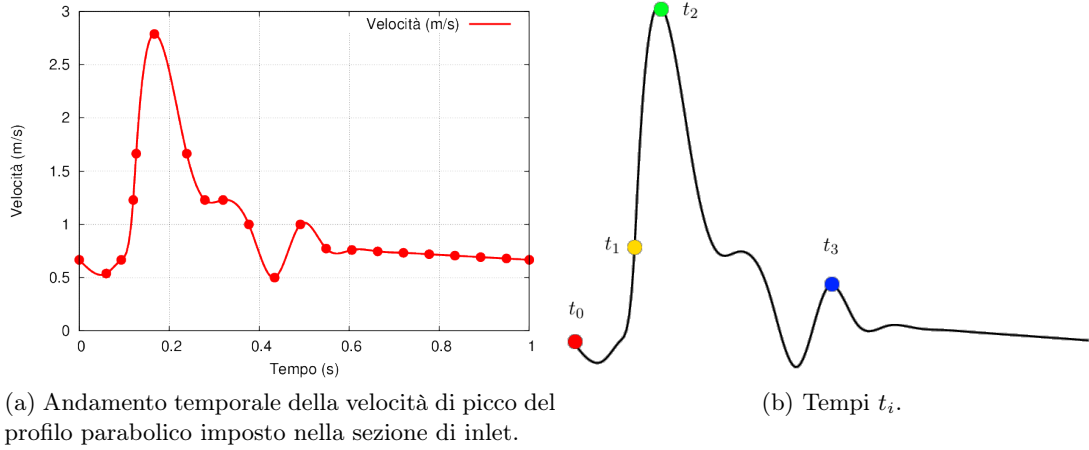


Figura 5.24: Velocità di picco per la sezione di inlet.

### 5.3.2 Problema di Stokes non stazionario

In questa sezione vengono presentati alcuni risultati ottenuti considerando un modello di Stokes non stazionario per il flusso del sangue, in cui la velocità  $\mathbf{u}(t) \in L^2(0, T; V_D(\Omega)) \cap C^0([0, T]; H(\Omega))$  e la pressione  $p(t) \in L^2(0, T; Q(\Omega))$  sono soluzione del seguente sistema:

$$\begin{cases} \partial_t \mathbf{u}(\Omega) - \nu \Delta \mathbf{u}(\Omega) + \nabla p(\Omega) = \mathbf{0}, & \text{in } \Omega \times (0, T), \\ \operatorname{div} \mathbf{u}(\Omega) = 0, & \text{in } \Omega \times (0, T), \\ \mathbf{u}(\Omega) = \mathbf{u}_{in}(t), & \text{su } \Gamma_{in} \times (0, T), \\ \mathbf{u}(\Omega) = \mathbf{0}, & \text{su } \Gamma_c \times (0, T), \\ \nu \frac{\partial \mathbf{u}(\Omega)}{\partial \mathbf{n}} - p(\Omega) \mathbf{n} = \mathbf{0}, & \text{su } \Gamma_{out} \times (0, T), \\ \mathbf{u}(\Omega) = \mathbf{u}_0(\Omega), & \text{in } \Omega \times \{0\}. \end{cases}$$

La condizione imposta all'inlet è un profilo parabolico  $\mathbf{u}_{in}(t) \in L^2(0, T; [H^{1/2}(\Gamma_{in})]^3)$ , il cui andamento temporale (mostrato in Figura 5.24(a)<sup>10</sup>) è stato ottenuto mediante interpolazione con una *spline cubica naturale* dei valori della velocità di picco in alcuni istanti temporali particolarmente rilevanti, indicati in [HNF<sup>+</sup>99]. Il dato iniziale  $\mathbf{u}_0(\Omega)$  è la soluzione del problema di Stokes stazionario con velocità di inlet data da  $\mathbf{u}_{in}(0)$ .

Considerando il seguente insieme di forme ammissibili

$$\mathcal{O}_{ad} = \{\Omega \subset \mathbb{R}^3 : \Gamma_{in} \cup \Gamma_{out} \text{ fissato}\},$$

si intende minimizzare l'energia dissipata

$$J(\Omega) = \frac{\nu}{2} \int_0^T \int_{\Omega} \chi(\Omega) |\nabla \mathbf{u}(\Omega)|^2 dx,$$

dove  $\chi(\Omega)$ , a seconda dei casi, è pari a  $\chi(\Omega) = 1$  (osservazione sull'intero dominio  $\Omega$ ) oppure  $\chi(\Omega) = \mathbf{1}_v(\Omega)$ , essendo  $\mathbf{1}_v(\Omega)$  la funzione indicatrice della regione a valle della stenosi.

L'analisi del problema di ottimizzazione di forma

$$\hat{\Omega} = \arg \min_{\Omega \in \mathcal{O}_{ad}} J(\Omega),$$

<sup>10</sup>Questo grafico, e tutti i seguenti, sono adimensionalizzati rispetto alla velocità di picco media  $\bar{u}_{in}$ , che si assume pari a 0.144 m/s.

#	Gradi di libertà di ottimizzazione	Gradi di libertà di deformazione	Variazione funzionale costo	Numero di iterazioni
<i>e</i>	16	72	-6%	19
<i>f</i>	12	60	-6%	16

#	$\theta_1$	$\theta_2$	Variazione diametro A	Variazione diametro B	Variazione diametro C	Variazione distanza D
<i>e</i>	40°	60°	+10%	0%*	0%	+10%
<i>f</i>	55°	45°	0%	0%*	0%	-15%

Tabella 5.4: Risultati dell'ottimizzazione di forma di un bypass aorto-coronarico, con flusso non stazionario. La legenda delle distanze e degli angoli è mostrata in Figura 5.15(b). La presenza di un asterisco indica che tale variazione di diametro è stata ottenuta mediante un vincolo.

già affrontata nella Sezione 3.3 nel caso in cui l'osservazione è distribuita sull'intero dominio ( $\chi(\Omega) = 1$ ), può essere generalizzata anche al caso in cui si osservi un sottodominio di  $\Omega$  ( $\chi(\Omega) = \mathbb{1}_v(\Omega)$ )

Le simulazioni sono state eseguite sul cluster *idra*, con 8 processori, considerando la coppia di elementi finiti stabili  $\mathbb{P}_{bolla}^1 - \mathbb{P}^1$  su una mesh di circa 15.000 elementi di volume (Figura 5.15(c)), ed avanzamento in tempo con il metodo di Eulero implicito, intervallo  $(0, T)$  pari ad un periodo del profilo di inlet e passo temporale  $\Delta t$  pari ad  $1/20$  di tale periodo. Questo permette di ottenere tempi computazionali sufficientemente ridotti: ciascuna risoluzione del problema di stato o del problema aggiunto richiede circa 3 minuti, il calcolo del funzionale costo richiede circa 1 minuto ed il calcolo del gradiente del funzionale costo rispetto al singolo grado di libertà richiede circa 10 minuti.

La Tabella 5.4 riassume i risultati ottenuti per le seguenti configurazioni di punti controllo:

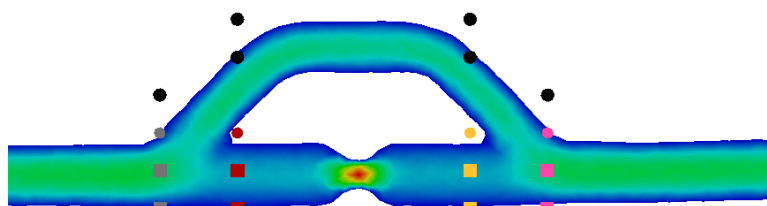
- (e) *Figura 5.25 - simulazione e* -  $\chi(\Omega) = 1$ : si considera la stessa configurazione della simulazione *c* del caso stazionario: 16 gradi di libertà di ottimizzazione, associati agli spostamenti in orizzontale ed in verticale dei sei punti di controllo indicati in Figura 5.25 con un cerchio nero ed allo spostamento in orizzontale dei quattro punti di controllo indicati con un cerchio colorato. Il valore dei gradi di libertà di ottimizzazione, sia orizzontali sia verticali, viene assegnato anche ai punti di controllo indicati con un quadrato dello stesso colore. Gli spostamenti così ottenuti sono replicati sulle sezioni parallele al piano longitudinale, per un totale di 72 gradi di libertà di deformazione. Infine, si osserva sull'intero dominio ( $\chi(\Omega) = 1$ ).
- (f) *Figura 5.26 - simulazione f* -  $\chi(\Omega) = \mathbb{1}_v(\Omega)$ : si considerano 12 gradi di libertà di ottimizzazione, associati agli spostamenti in orizzontale ed in verticale dei quattro punti di controllo indicati in Figura 5.26 con un cerchio nero ed allo spostamento in orizzontale dei quattro punti di controllo indicati con un cerchio colorato. Il valore dei gradi di libertà di ottimizzazione, sia orizzontali sia verticali, viene assegnato anche ai punti di controllo indicati con un quadrato dello stesso colore. Gli spostamenti così ottenuti sono replicati sulle sezioni parallele al piano longitudinale, per un totale di 60 gradi di libertà di deformazione. Infine, si osserva sulla regione a valle dell'occlusione ( $\chi(\Omega) = \mathbb{1}_v(\Omega)$ ).

In seguito vengono rappresentati i risultati in corrispondenza degli istanti temporali più caratteristici del fenomeno: con riferimento alla Figura 5.24(b), si definiscono gli istanti  $t_i$ ,  $i = 0, 1, 2, 3$ , dove  $t_0$  indica l'istante al termine della fase sistolica,  $t_1$  il tempo di massima accelerazione diastolica,  $t_2$  il tempo di massima velocità diastolica e  $t_3$  il tempo di massima velocità sistolica.

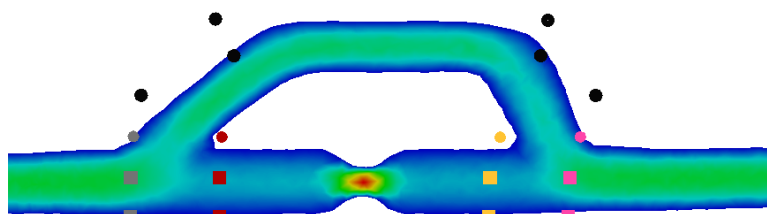
La Figura 5.27 mostra il modulo della velocità nella sezione longitudinale, agli istanti temporali  $t_i$ , per la configurazione indeformata e per le configurazioni ottime delle due simulazioni. Le variazioni di velocità nella coronaria e nel bypass sono governate dalla natura pulsatile del profilo imposto in inlet: la velocità è minima nell'istante  $t = t_0$  (posto alla fine della fase sistolica), rimane sostanzialmente invariata all'istante di massima accelerazione diastolica ( $t = t_1$ ) e concentra una elevata variazione nell'intervallo  $[t_1, t_2]$ <sup>11</sup>, al termine del quale la velocità è triplicata rispetto al tempo iniziale  $t = t_0$ . La velocità diminuisce nel resto della fase diastolica, raggiungendo un nuovo massimo locale in  $t = t_3$  (tempo di massima velocità sistolica). Analogamente al caso stazionario, il flusso imposto dal profilo di inlet si divide, nella regione dell'innesto prossimale, tra il *graft* e l'arteria occlusa, come evidenziato dalle *streamlines* (Figura 5.29); la velocità del sangue che percorre il *graft* è più elevata di quella del flusso che prosegue nella coronaria. La massima velocità, a tempo  $t_i$  fissato, è raggiunta nella zona della stenosi: la forma ottima della simulazione  $e$  (per ogni  $i$ ) e la forma ottima della simulazione  $f$  (per  $i \neq 2$ ) portano a diminuire la velocità massima rispetto alla configurazione indeformata, grazie ad un miglior flusso nel *graft*. I profili di velocità al variare delle sezioni (Figura 5.28) mostrano che, a differenza del caso stazionario, la velocità non assume un profilo parabolico al tempo  $t = t_2$ , ma diventa costante a sufficiente distanza dalla parete; questa osservazione è in accordo con i risultati sperimentali e computazionali mostrati in [DOYM11, BDFD01, LFB08].

La Figura 5.30 mostra l'andamento dello sforzo di taglio a parete (WSS) nella regione a valle della stenosi; si confermano le stesse conclusioni del caso stazionario: i risultati manifestano la presenza di una regione di basso WSS a monte dell'innesto (con possibilità della completa ostruzione della coronaria) e di una zona di alto WSS a valle dell'innesto (il cui studio verrà approfondito nella Sezione 5.3.3 con la validazione mediante equazioni di Navier-Stokes), e ribadiscono la necessità di mantenere l'angolo  $\theta_2$  di Figura 5.15(b) sufficientemente piccolo al fine di evitare la comparsa di zone di basso WSS sulla parte terminale del *graft*.

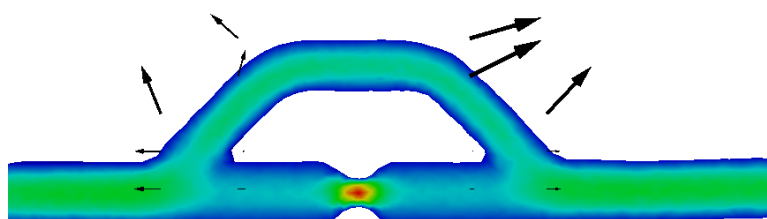
<sup>11</sup>Questo comportamento sembra essere causato dal passo temporale troppo ampio: infatti, nella Sezione 5.3.3 si considera una discretizzazione temporale più fine ed una variazione di velocità non trascurabile rispetto a  $t = t_0$  è già presente nell'istante di massima accelerazione diastolica.



(a) Configurazione indeformata (sezione).

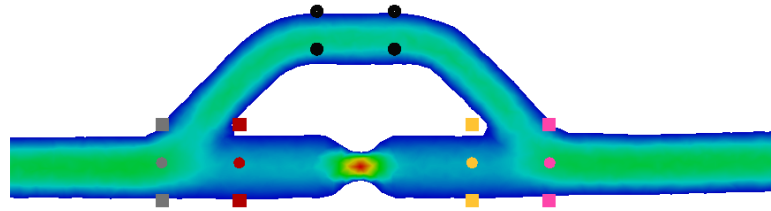


(b) Configurazione ottima (sezione).

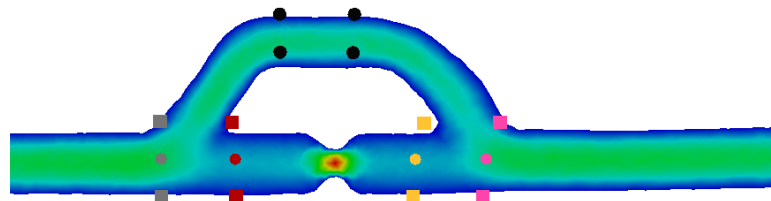


(c) Spostamento imposto sui punti di controllo abilitati per ottenere la configurazione ottima (sezione).

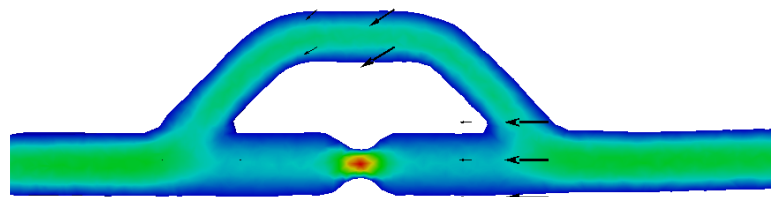
Figura 5.25: Simulazione *e*: rappresentazione della configurazione ottima, dei punti di controllo con gradi di libertà di ottimizzazione (indicati con un cerchio) e dei punti di controllo con gradi di libertà di deformazione (unione dei precedenti con i punti di controllo indicati con un quadrato, da replicarsi anche per le altre due sezioni parallele). I gradi di libertà di deformazione sono legati ai gradi di libertà di ottimizzazione del medesimo colore.



(a) Configurazione indeformata (sezione).

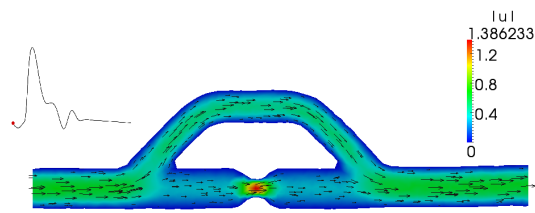


(b) Configurazione ottima (sezione).

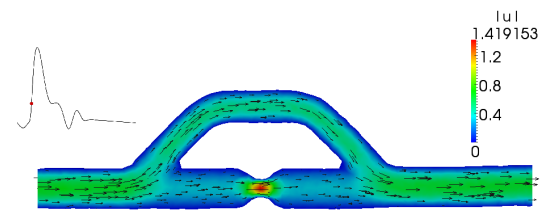


(c) Spostamento imposto sui punti di controllo abilitati per ottenere la configurazione ottima (sezione).

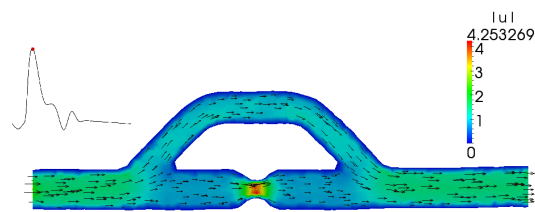
Figura 5.26: Simulazione  $f$ : rappresentazione della configurazione ottima, dei punti di controllo con gradi di libertà di ottimizzazione (indicati con un cerchio) e dei punti di controllo con gradi di libertà di deformazione (unione dei precedenti con i punti di controllo indicati con un quadrato, da replicarsi anche per le altre due sezioni parallele). I gradi di libertà di deformazione sono legati ai gradi di libertà di ottimizzazione del medesimo colore.



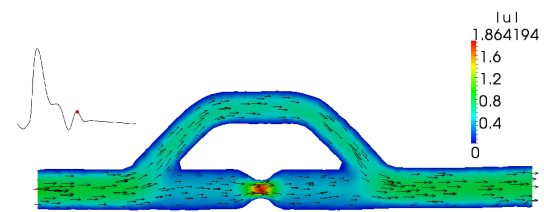
(a) Configurazione indeformata: modulo della velocità al tempo  $t = t_0$ .



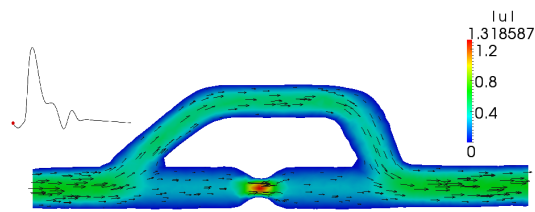
(b) Configurazione indeformata: modulo della velocità al tempo  $t = t_1$ .



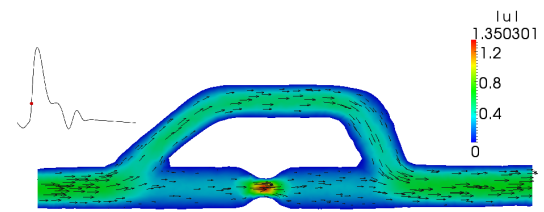
(c) Configurazione indeformata: modulo della velocità al tempo  $t = t_2$ .



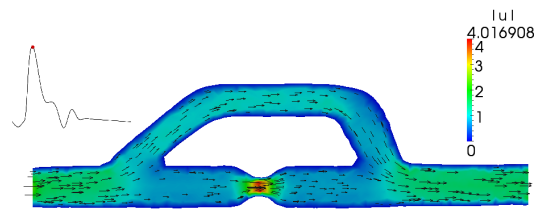
(d) Configurazione indeformata: modulo della velocità al tempo  $t = t_3$ .



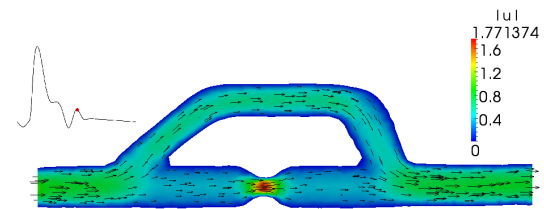
(e) Configurazione ottima, simulazione *e*: modulo della velocità al tempo  $t = t_0$ .



(f) Configurazione ottima, simulazione *e*: modulo della velocità al tempo  $t = t_1$ .



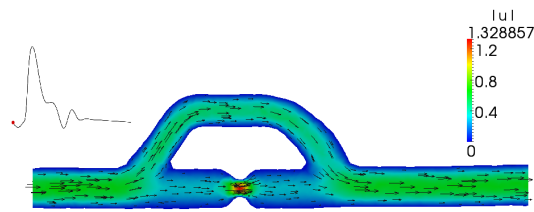
(g) Configurazione ottima, simulazione *e*: modulo della velocità al tempo  $t = t_2$ .



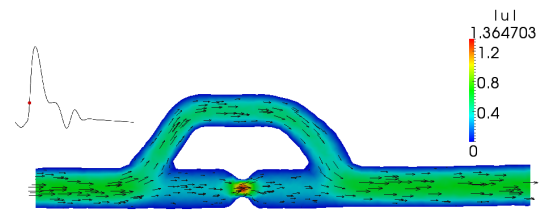
(h) Configurazione ottima, simulazione *e*: modulo della velocità al tempo  $t = t_3$ .

Figura 5.27: Velocità nella sezione longitudinale a diversi istanti temporali.

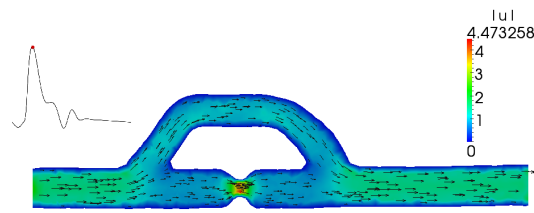




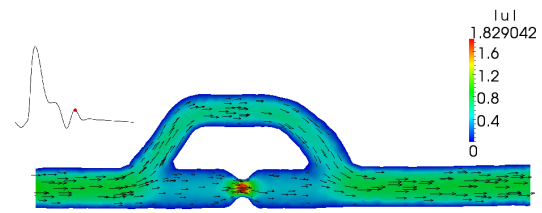
(i) Configurazione ottima, simulazione  $f$ : modulo della velocità al tempo  $t = t_0$ .



(j) Configurazione ottima, simulazione  $f$ : modulo della velocità al tempo  $t = t_1$ .

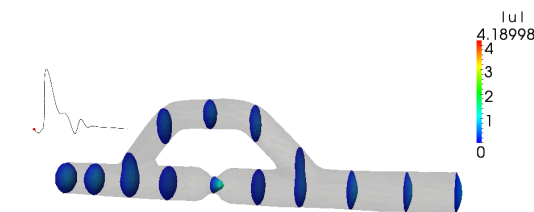


(k) Configurazione ottima, simulazione  $f$ : modulo della velocità al tempo  $t = t_2$ .

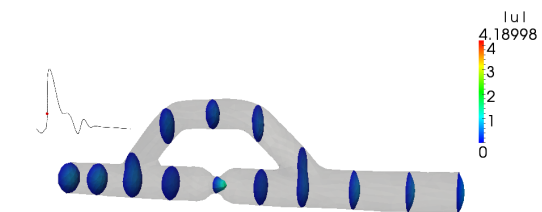


(l) Configurazione ottima, simulazione  $f$ : modulo della velocità al tempo  $t = t_3$ .

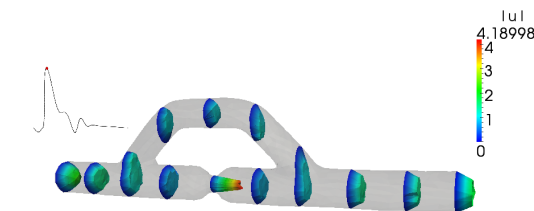
Figura 5.27: Velocità nella sezione longitudinale a diversi istanti temporali (2).



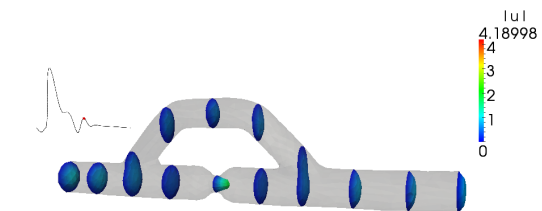
(a) Configurazione indeformata: profili di velocità su varie sezioni al tempo  $t = t_0$ .



(b) Configurazione indeformata: profili di velocità su varie sezioni al tempo  $t = t_1$ .

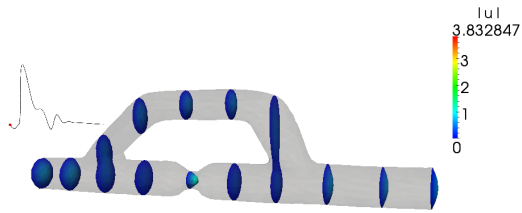


(c) Configurazione indeformata: profili di velocità su varie sezioni al tempo  $t = t_2$ .

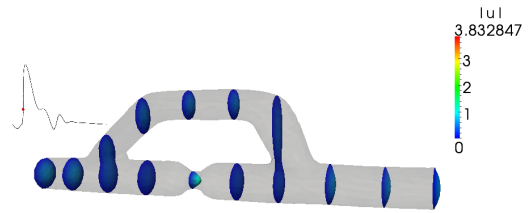


(d) Configurazione indeformata: profili di velocità su varie sezioni al tempo  $t = t_3$ .

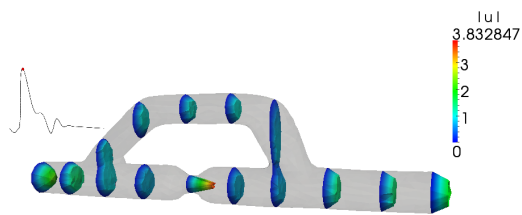
Figura 5.28: Profili di velocità su varie sezioni a diversi istanti temporali.



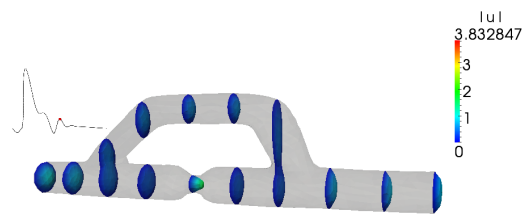
(e) Configurazione ottima, simulazione *e*: profili di velocità su varie sezioni al tempo  $t = t_0$ .



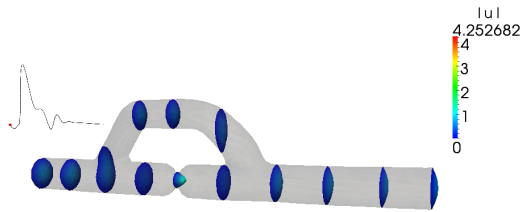
(f) Configurazione ottima, simulazione *e*: profili di velocità su varie sezioni al tempo  $t = t_1$ .



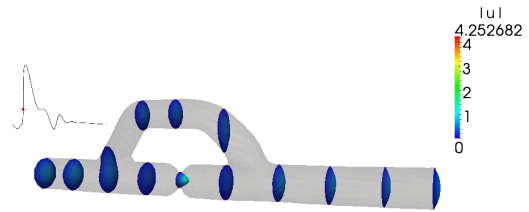
(g) Configurazione ottima, simulazione *e*: profili di velocità su varie sezioni al tempo  $t = t_2$ .



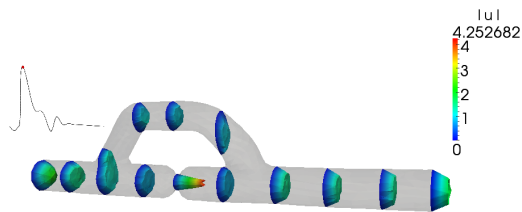
(h) Configurazione ottima, simulazione *e*: profili di velocità su varie sezioni al tempo  $t = t_3$ .



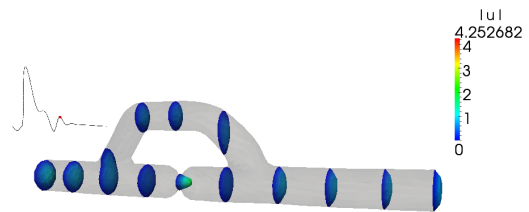
(i) Configurazione ottima, simulazione *f*: profili di velocità su varie sezioni al tempo  $t = t_0$ .



(j) Configurazione ottima, simulazione *f*: profili di velocità su varie sezioni al tempo  $t = t_1$ .



(k) Configurazione ottima, simulazione *f*: profili di velocità su varie sezioni al tempo  $t = t_2$ .



(l) Configurazione ottima, simulazione *f*: profili di velocità su varie sezioni al tempo  $t = t_3$ .

Figura 5.28: Profili di velocità su varie sezioni a diversi istanti temporali (2).

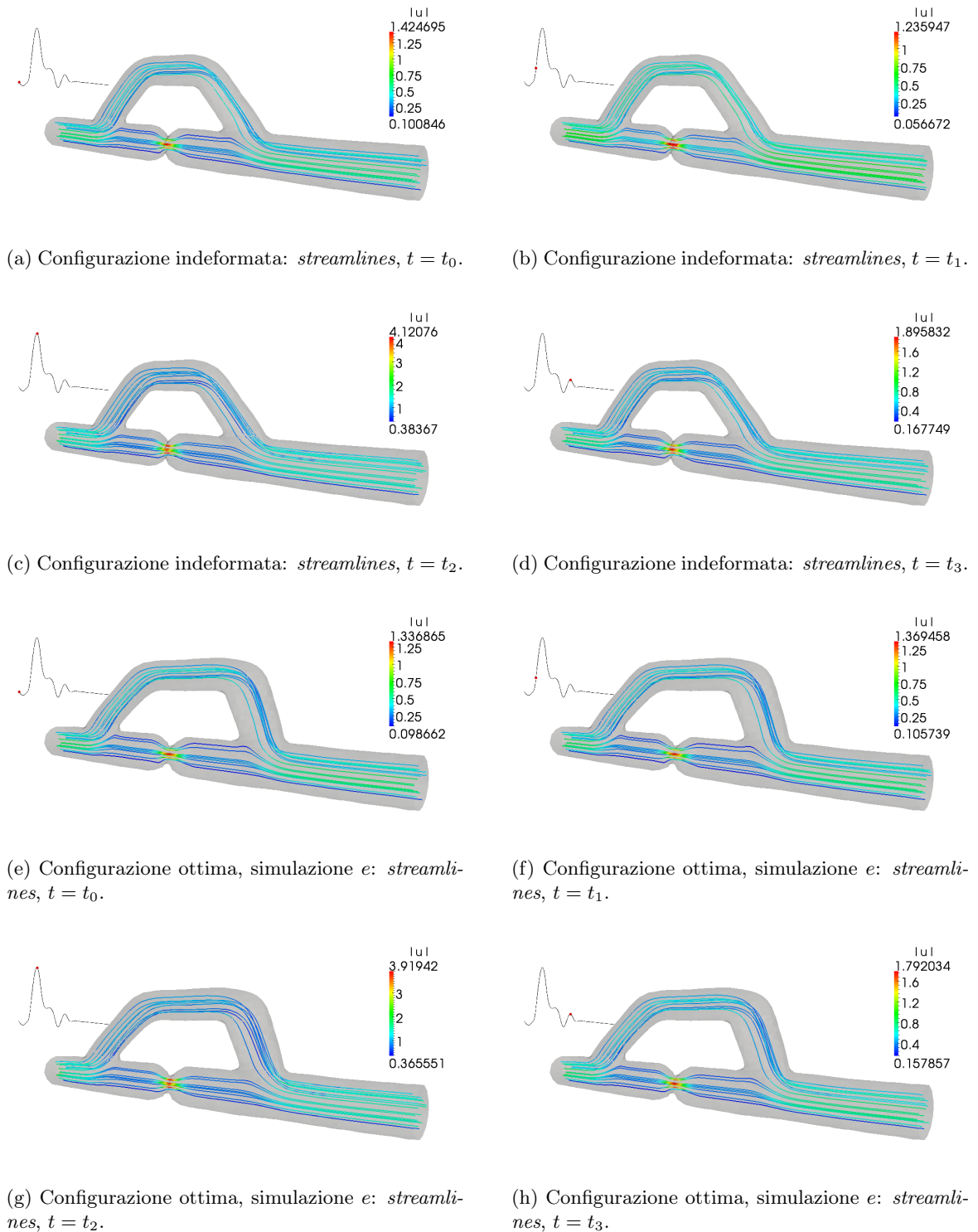


Figura 5.29: *Streamlines* a diversi istanti temporali.

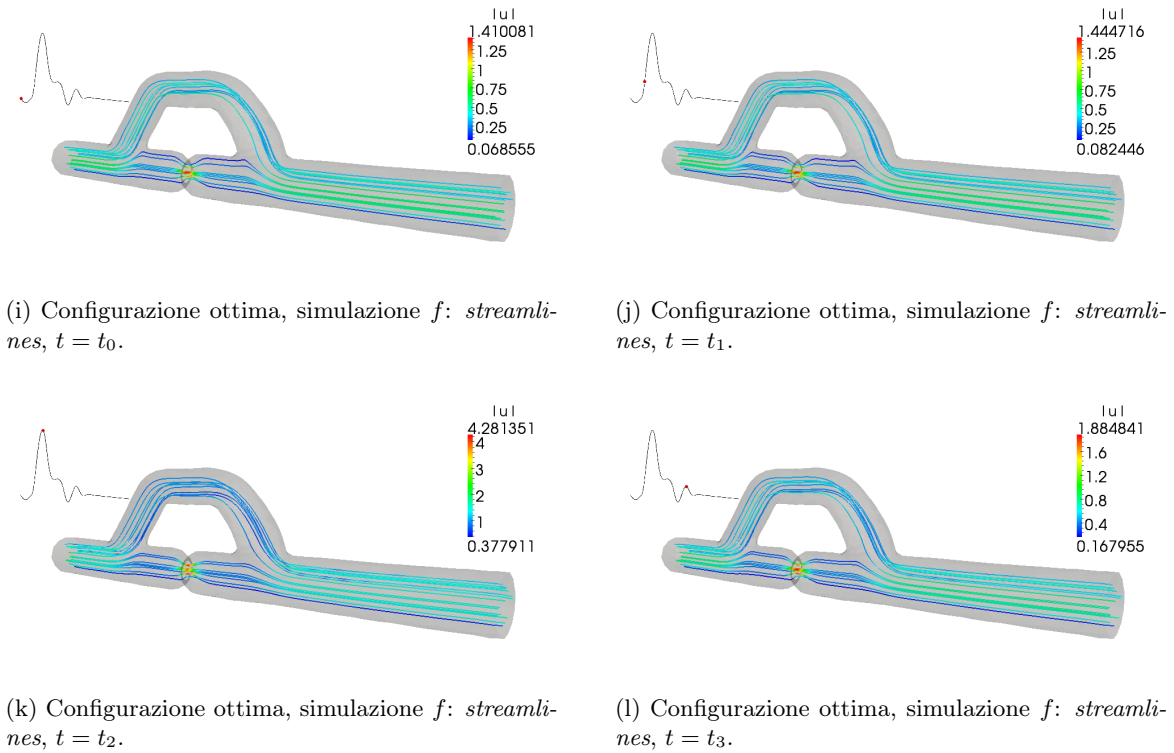


Figura 5.29: *Streamlines* a diversi istanti temporali (2).

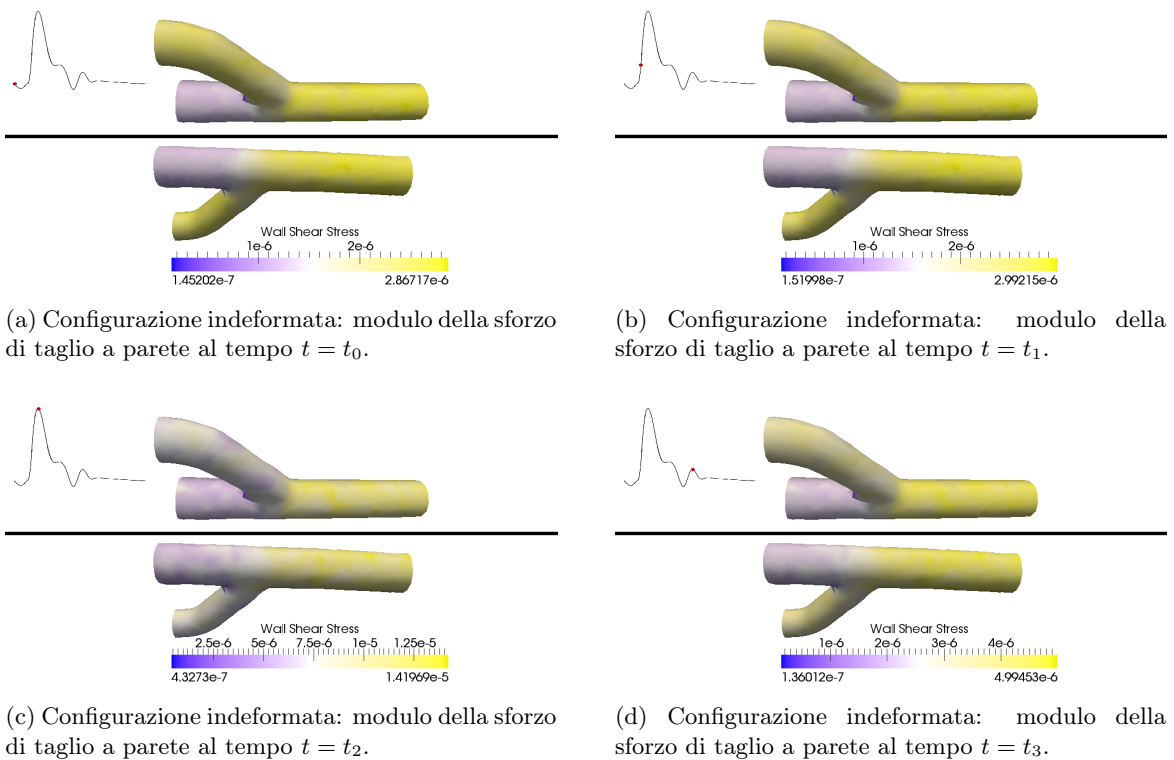
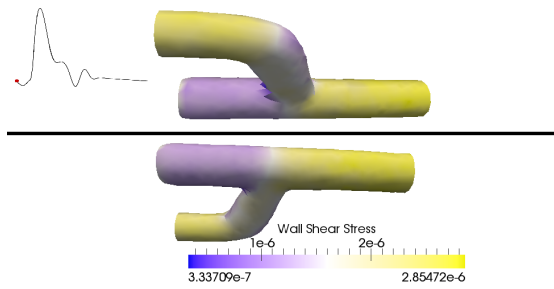
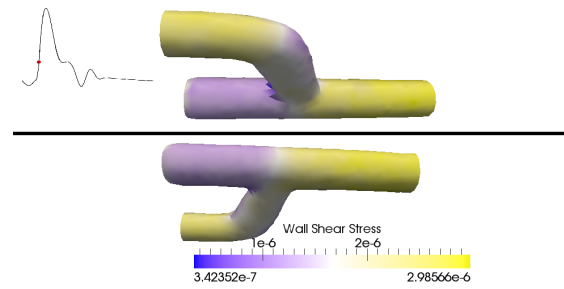


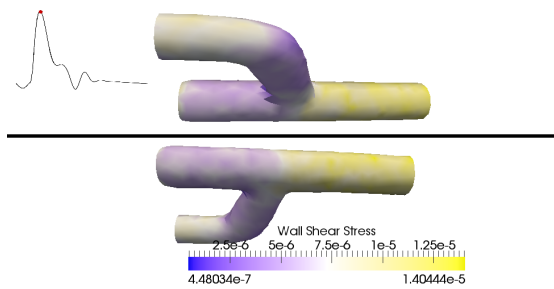
Figura 5.30: Modulo della sforzo di taglio a parete nella regione a valle della stenosi, a diversi istanti temporali.



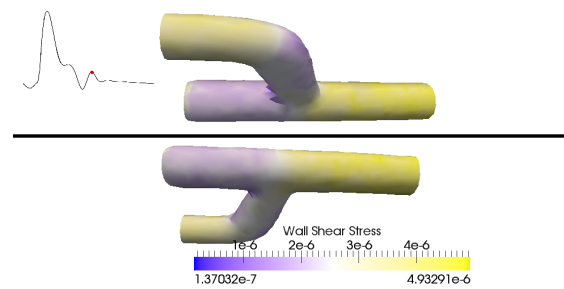
(e) Configurazione ottima, simulazione *e*: modulo della sforzo di taglio a parete al tempo  $t = t_0$ .



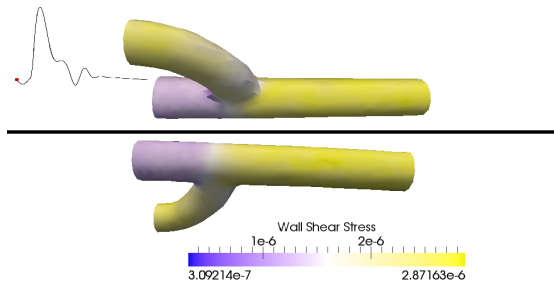
(f) Configurazione ottima, simulazione *e*: modulo della sforzo di taglio a parete al tempo  $t = t_1$ .



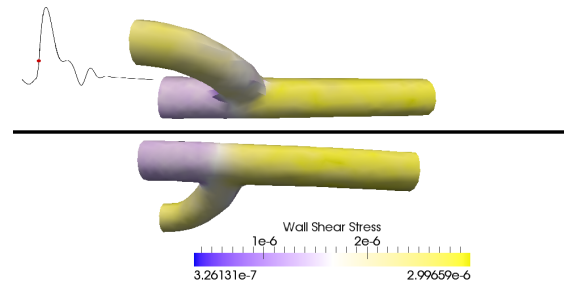
(g) Configurazione ottima, simulazione *e*: modulo della sforzo di taglio a parete al tempo  $t = t_2$ .



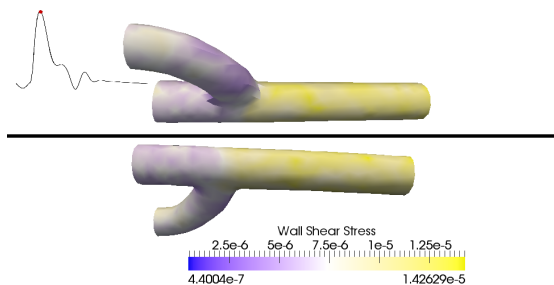
(h) Configurazione ottima, simulazione *e*: modulo della sforzo di taglio a parete al tempo  $t = t_3$ .



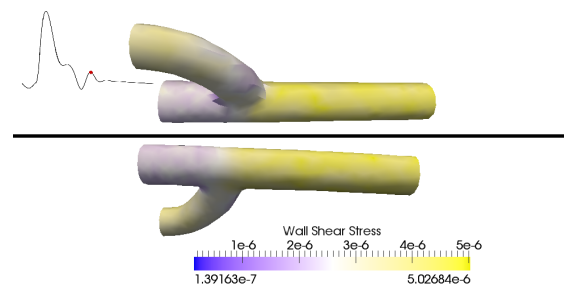
(i) Configurazione ottima, simulazione *f*: modulo della sforzo di taglio a parete al tempo  $t = t_0$ .



(j) Configurazione ottima, simulazione *f*: modulo della sforzo di taglio a parete al tempo  $t = t_1$ .



(k) Configurazione ottima, simulazione *f*: modulo della sforzo di taglio a parete al tempo  $t = t_2$ .



(l) Configurazione ottima, simulazione *f*: modulo della sforzo di taglio a parete al tempo  $t = t_3$ .

Figura 5.30: Modulo della sforzo di taglio a parete nella regione a valle della stenosi, a diversi istanti temporali (2).

### 5.3.3 Validazione con problema di Navier-Stokes non stazionario

Al fine di confrontare i risultati ottenuti nelle sezioni precedenti con quelli citati nella Sezione 1.3, in particolare per l'analisi dello sforzo di taglio a parete e dell'*oscillating shear index*, è stata eseguita una validazione simulando in ciascuna forma ottima un flusso non stazionario governato dal seguente problema di Navier-Stokes: trovare  $\mathbf{u}(t) \in L^2(0, T; V_D(\Omega)) \cap C^0([0, T]; H(\Omega))$  e  $p(t) \in L^2(0, T; Q(\Omega))$  tali che

$$\begin{cases} \partial_t \mathbf{u}(\Omega) - \nu \Delta \mathbf{u}(\Omega) + (\mathbf{u}(\Omega) \cdot \nabla) \mathbf{u}(\Omega) + \nabla p(\Omega) = \mathbf{0}, & \text{in } \Omega \times (0, T), \\ \operatorname{div} \mathbf{u}(\Omega) = 0, & \text{in } \Omega \times (0, T), \\ \mathbf{u}(\Omega) = \mathbf{u}_{in}(t), & \text{su } \Gamma_{in} \times (0, T), \\ \mathbf{u}(\Omega) = \mathbf{0}, & \text{su } \Gamma_c \times (0, T), \\ \nu \frac{\partial \mathbf{u}(\Omega)}{\partial \mathbf{n}} - p(\Omega) \mathbf{n} = \mathbf{0}, & \text{su } \Gamma_{out} \times (0, T), \\ \mathbf{u}(\Omega) = \mathbf{u}_0(\Omega), & \text{in } \Omega \times \{0\}. \end{cases}$$

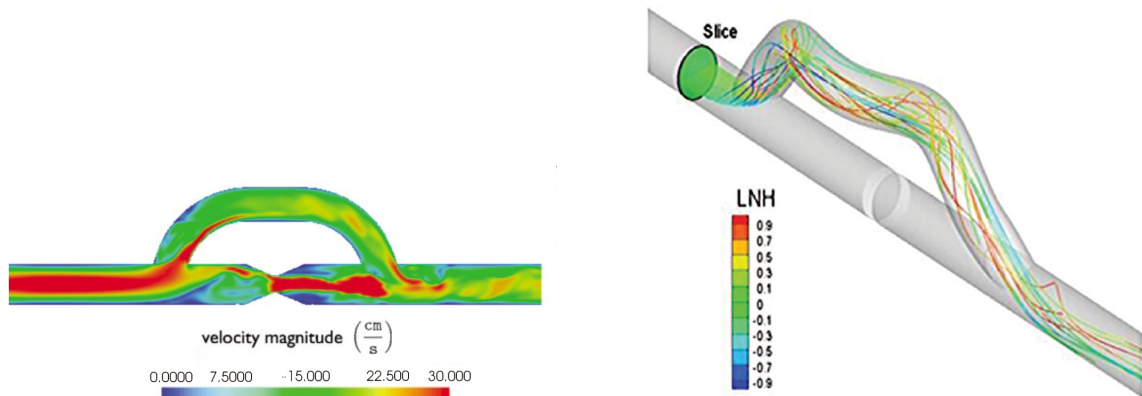
La condizione di inlet  $\mathbf{u}_{in}(t) \in L^2(0, T; [H^{1/2}(\Gamma_{in})]^3)$  è un profilo parabolico, il cui andamento temporale è mostrato in Figura 5.24(a)<sup>12</sup>. La velocità di picco media  $\bar{u}_{in}$  è scelta in modo che il numero di Reynolds, basato sul diametro della sezione di inlet, sia pari a 260 (dato tratto da [WZJ<sup>+</sup>11]); il massimo numero di Reynolds, al variare del tempo, è raggiunto in  $t = t_2$  (velocità diastolica massima) e vale 725. Il dato iniziale  $\mathbf{u}_0(\Omega)$  è soluzione del problema di Navier-Stokes stazionario con velocità di inlet data da  $\mathbf{u}_{in}(0)$ .

Le simulazioni sono state eseguite sul cluster *idra*, con 8 processori, considerando una discretizzazione spaziale e temporale più fine rispetto a quelle impiegate nelle precedenti sezioni, in modo da cogliere più accuratamente il comportamento non lineare: è stata impiegata la coppia di elementi finiti stabili  $\mathbb{P}_{bolla}^1 - \mathbb{P}^1$  su una mesh di circa 70.000 elementi di volume (Figura 5.15(d)), ed un avanzamento in tempo con il metodo di Eulero implicito. Per evitare effetti dovuti al transitorio iniziale l'intervallo  $(0, T)$  è stato scelto pari a dieci volte il periodo del profilo di inlet; il passo temporale  $\Delta t$  è pari ad  $1/100$  di tale periodo. Nell'approssimazione del problema di Navier-Stokes stazionario la non linearità è stata risolta con un metodo di Newton; nell'approssimazione del problema non stazionario è stato invece utilizzato un trattamento semi-implicito del termine convettivo. La simulazione ha richiesto circa 2 ore.

La Figura 5.32 mostra il modulo della velocità nella sezione longitudinale, agli istanti temporali  $t_i$ , per la configurazione indeformata e per le configurazioni ottime ottenute con il modello semplificato di Stokes (stazionario e non stazionario); la Figura 5.33 rappresenta i profili di velocità su alcune sezioni. Come nel caso stazionario, le variazioni di velocità nella coronaria e nel bypass sono governate dalla natura pulsatile del profilo di inlet: si osserva velocità minima in  $t = t_0$  (tempo in cui termina la sistole), velocità crescente in  $t = t_1$  (tempo di massima accelerazione diastolica) fino a raggiungere il massimo in  $t = t_2$  (tempo di massima velocità diastolica del profilo di inlet); la velocità diminuisce nel resto della fase diastolica, con un nuovo massimo locale in  $t = t_3$  (tempo di massima velocità sistolica del profilo di inlet). Confrontando i risultati della Figura 5.32 con la Figura 5.31(a), che mostra il modulo della velocità per una geometria simile ma con una discretizzazione molto più fine (con mesh composta da circa  $10^6$  elementi di volume e passo temporale  $\Delta t$  pari ad  $1/2.000$  del periodo del profilo di inlet; tratta da [SM10]), si può notare che la natura non lineare del problema di Navier-Stokes è particolarmente evidente nelle seguenti tre regioni:

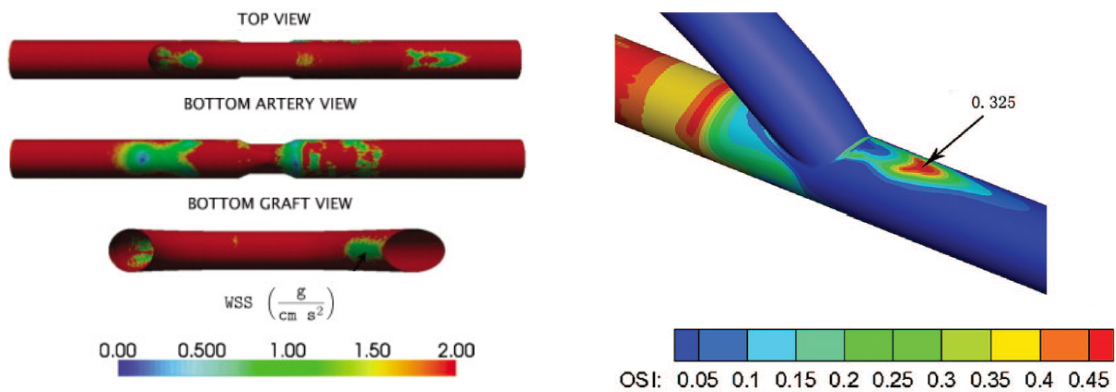
- nel tratto di coronaria compreso tra innesto prossimale e la stenosi, in cui il profilo di velocità si discosta notevolmente da quello simmetrico previsto dal modello di Stokes: si riscontra infatti in questa regione una zona di elevata velocità nei pressi della parete superiore

<sup>12</sup>Questo grafico, e tutti i seguenti, sono adimensionalizzati rispetto alla velocità di picco media  $\bar{u}_{in}$ .



(a) Modulo della velocità: nelle simulazioni in cui si considera il modello emodinamico completo (problema di Navier-Stokes) la natura non lineare dell'equazione è particolarmente evidente intorno agli innesti e nella zona a valle della stenosi. Tratto da [SM10].

(b) Le *streamlines* di un *graft* di tipo elicoidale mostrano un comportamento migliorato nell'innesto distale. Tratto da [WZJ<sup>+</sup>11].



(c) Andamento degli sforzi di taglio a parete. Tratto da [SM10].

(d) Andamento dell'OSI. Tratto da [WZJ<sup>+</sup>11].

Figura 5.31: Confronto con alcuni risultati tratti da [WZJ<sup>+</sup>11, SM10].

dell'arteria, ed una zona di basse velocità vicino alla parete inferiore della coronaria. In [SM10] viene evidenziato come quest'ultima regione sia anche una zona di ricircolo, con la formazione di un vortice particolarmente accentuato nella fase di sistole. L'asimmetria del profilo di velocità è ben evidenziata anche nelle simulazioni presentate in questa sezione, che tuttavia, a causa della discretizzazione più lasca, non colgono la formazione del vortice;

- nel tratto di coronaria compreso tra la stenosi e l'innesto distale, in cui si forma un getto ad elevata velocità causato dalla stenosi (*post-stenotic jet*). Al contrario, fissato il tempo  $t = t_i$ , nel modello di Stokes si osservava la massima velocità solo al centro della stenosi. La formazione del getto è confermata anche nei risultati presentati in questa sezione, nonostante le sue dimensioni siano notevolmente ridotte rispetto a quelle di Figura 5.31(a). In [SM10] viene evidenziato come l'interazione tra un getto particolarmente elongato ed il flusso proveniente dall'innesto distale sia causa, in prossimità della parete superiore della coronaria, della formazione di una zona di ricircolo;
- nel tratto di coronaria a valle dell'innesto distale: la confluenza dal *graft* nella coronaria comporta la formazione di una regione di bassa velocità in prossimità della parete superiore dell'arteria, a differenza del profilo simmetrico fornito dalle equazioni di Stokes. La simmetria del profilo di velocità per il flusso di Stokes era causa, nei risultati presentati nelle sezioni precedenti, del valore costante dello sforzo di taglio a parete nella regione a valle dell'innesto distale.

Le *streamlines* agli istanti temporali  $t_i$  sono raffigurate nella Figura 5.34, per la configurazione indeformata e per le configurazioni ottime ottenute con il modello semplificato di Stokes. Le *streamlines* sono colorate secondo l'indice  $LNH$  (*local normalized helicity*) dato da:

$$LNH = \frac{\mathbf{u} \cdot (\nabla \times \mathbf{u})}{|\mathbf{u}| |\nabla \times \mathbf{u}|}, \quad -1 \leq LNH \leq 1;$$

un valore nullo di  $LNH$  indica un moto puramente assiale o circonferenziale, mentre un valore di modulo unitario indica un moto puramente elicoidale. Questo indice è particolarmente importante nel confronto tra la geometria tradizionale ed una di tipo elicoidale, in cui si considera un bypass "a forma di M": infatti è stato mostrato (si veda ad esempio [WZJ<sup>+</sup>11]) che la presenza di valori elevati di  $|LNH|$  nella zona centrale del *graft* favorisce un rimescolamento del fluido a valle dell'anastomosi distale, in particolare nella regione in prossimità della parete superiore dell'arteria, aumentandone sia la velocità sia il WSS e contribuendo dunque a diminuire l'entità del processo di ristenosi (Figura 5.31(b)). Coerentemente con i risultati di [WZJ<sup>+</sup>11], la Figura 5.34 mostra che in una geometria tradizionale la zona centrale del *graft* non contribuisce al rimescolamento ( $LNH$  nullo) e che si ottengono valori elevati di  $LNH$  solamente in prossimità degli innesti.

La vorticità per la configurazione indeformata e per le configurazioni ottime per il modello di Stokes è rappresentata nella Figura 5.35: si nota la presenza di zone di elevata vorticità in prossimità della linea di sutura distale e lungo tutta la parete inferiore dell'arteria, a valle dell'innesto distale. Queste zone sono anche quelle particolarmente soggette a ristenosi: per questo motivo, seguendo [QR03, Roz05a, AQR07], è stato legittimo considerare il problema di ottimizzazione di forma con minimizzazione di un funzionale costo dipendente dalla vorticità (nello studio in esame è stata in realtà minimizzata l'energia dissipata che, come visto nella Sezione 3.1, fornisce un funzionale costo equivalente sotto opportune ipotesi).

Le distribuzioni dello sforzo di taglio a parete a valle della stenosi sono rappresentate nella Figura 5.36; a differenza delle simulazioni con equazioni di Stokes, questi risultati mettono in evidenza le variazioni del WSS nella zona a valle dell'innesto (solitamente sede di ristenosi), soprattutto ai tempi  $t = t_0$  e  $t = t_3$ . In particolare, la posizione delle regioni di basso WSS è in buon accordo con quanto mostrato nella Figura 5.31(c). Confrontando la distribuzione dello sforzo di taglio a parete al variare della forma, si possono trarre le seguenti conclusioni sull'*optimal design* di un bypass aorto-coronarico:

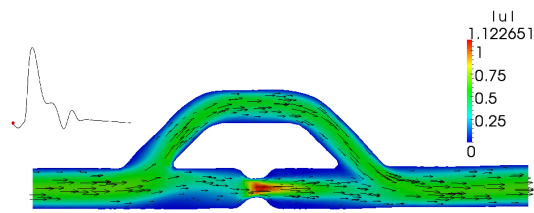


1. sono preferibili geometrie con angoli  $\theta_2$  inferiori a  $55^\circ$ : infatti le configurazioni ottime delle simulazioni  $b$ ,  $d'$  ed  $f$  mostrano una zona ridotta di basso WSS a valle dell'innesto (zona in viola chiaro nella vista dall'alto). L'angolo d'innesto troppo elevato riscontrato nelle configurazioni ottime delle simulazioni  $a$  ed  $e$  causa zone di basso WSS anche nell'innesto; nel caso della simulazione  $d''$  l'innesto quasi ortogonale all'arteria causa una regione piuttosto estesa di basso WSS a valle dall'innesto. Nonostante si abbia  $\theta_2$  pari a  $55^\circ$ , un WSS molto basso è presente anche nella configurazione ottima della simulazione  $c$ : ciò è causato non tanto dell'ampiezza di  $\theta_2$ , ma piuttosto del rigonfiamento innaturale presente nel *graft*. Queste considerazioni sono anche confermate dai risultati citati in [DBB01, BD00].
2. sono preferibili geometrie in cui l'innesto è suturato ad una maggiore distanza dall'occlusione: infatti le configurazioni ottime delle simulazioni  $b$  ed  $e$  presentano un valore minimo di WSS più elevato di quello della configurazione di riferimento. Ciò è confermato anche dalla configurazione ottima della simulazione  $c$ , in cui tuttavia il valore massimo di WSS è sensibilmente ridotto a causa del rigonfiamento. Al contrario, le configurazioni ottime della simulazione  $a$  (distanza dall'occlusione sostanzialmente invariata rispetto alla configurazione indeformata) e della simulazione  $f$  (distanza dall'occlusione minore rispetto alla configurazione indeformata) presentano un valore minimo di WSS molto basso. Appare più complessa l'analisi delle simulazioni  $d'$  e  $d''$ , per le quali si sospetta che l'eccessiva deformazione della stenosi<sup>13</sup> abbia portato a valori minimi di sforzi di taglio artificialmente alti. Queste considerazioni sono anche confermate dai risultati di [BDFD01, BD00].

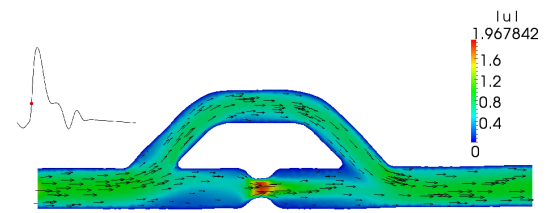
Infine, la Figura 5.37 mostra la valutazione dell'*oscillating shear index* (OSI): come previsto, si ottengono zone di elevato OSI (possibili regioni di ristrenosi) a valle dell'occlusione, sulla parete immediatamente a valle dell'innesto e, in particolare nelle configurazioni con angoli  $\theta_2$  elevati, anche nel *graft*. La forte natura oscillante del flusso in queste regioni è confermata anche dai risultati di [WZJ<sup>+</sup>11]; tuttavia la Figura 5.37 mostra un valore massimo di OSI notevolmente inferiore a quanto mostrato in Figura 5.31(d): si ritiene che ciò sia causato dalla mancata interazione tra il *post-stenotic jet* ed il flusso proveniente dall'innesto distale. Dunque, come discusso in precedenza, l'utilizzo di una mesh più fine, che catturi la non linearità del flusso, risulta necessario per una descrizione più accurata della natura oscillatoria del flusso.

---

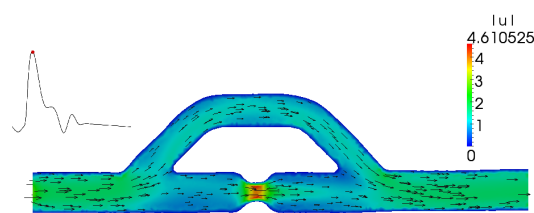
<sup>13</sup>Si confrontino, ad esempio, le sottfigure (b) e (c) con (a) in Figura 5.19.



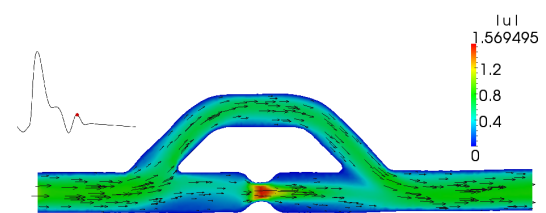
(a) Configurazione indeformata: modulo della velocità al tempo  $t = t_0$ .



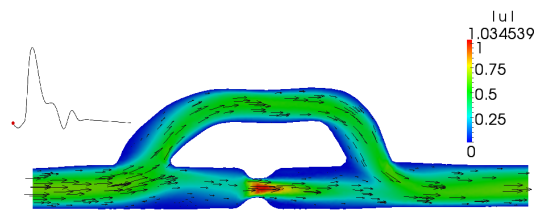
(b) Configurazione indeformata: modulo della velocità al tempo  $t = t_1$ .



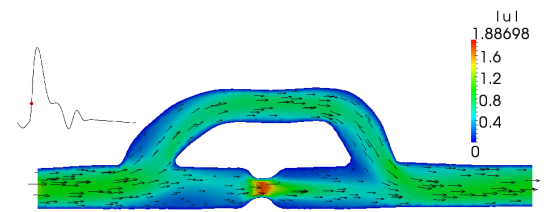
(c) Configurazione indeformata: modulo della velocità al tempo  $t = t_2$ .



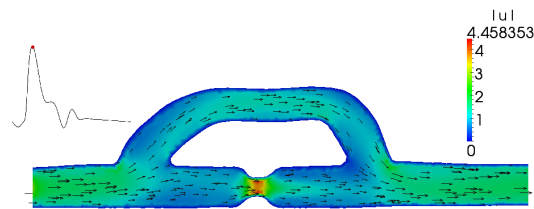
(d) Configurazione indeformata: modulo della velocità al tempo  $t = t_3$ .



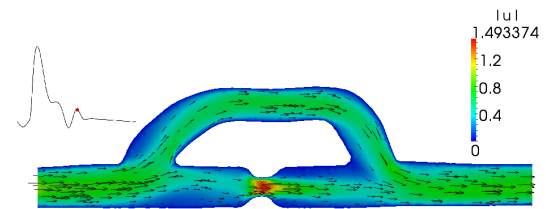
(e) Configurazione ottima, simulazione *a*: modulo della velocità al tempo  $t = t_0$ .



(f) Configurazione ottima, simulazione *a*: modulo della velocità al tempo  $t = t_1$ .

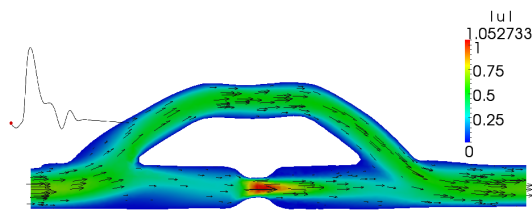


(g) Configurazione ottima, simulazione *a*: modulo della velocità al tempo  $t = t_2$ .

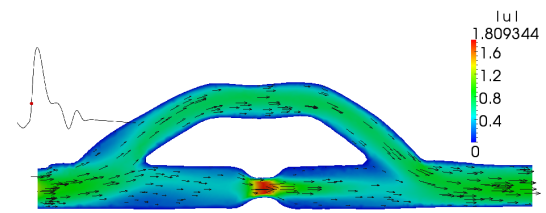


(h) Configurazione ottima, simulazione *a*: modulo della velocità al tempo  $t = t_3$ .

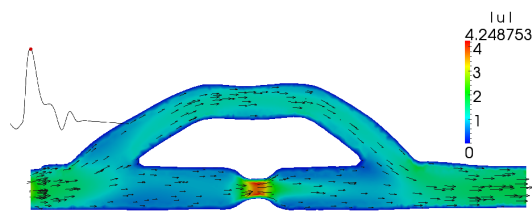
Figura 5.32: Velocità nella sezione longitudinale a diversi istanti temporali.



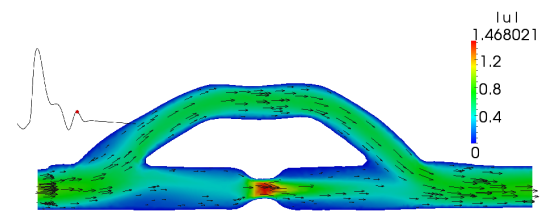
(i) Configurazione ottima, simulazione *b*: modulo della velocità al tempo  $t = t_0$ .



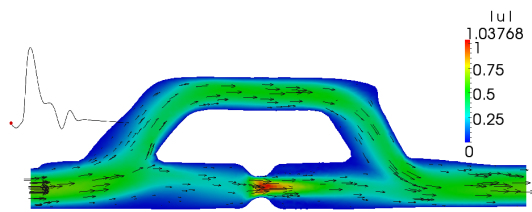
(j) Configurazione ottima, simulazione *b*: modulo della velocità al tempo  $t = t_1$ .



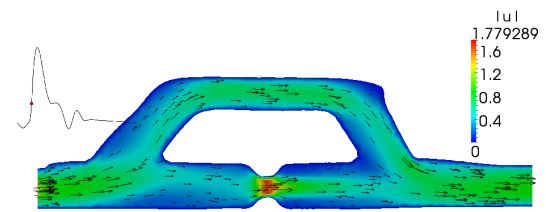
(k) Configurazione ottima, simulazione *b*: modulo della velocità al tempo  $t = t_2$ .



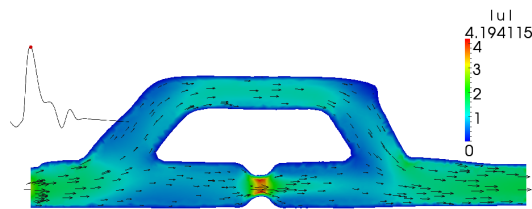
(l) Configurazione ottima, simulazione *b*: modulo della velocità al tempo  $t = t_3$ .



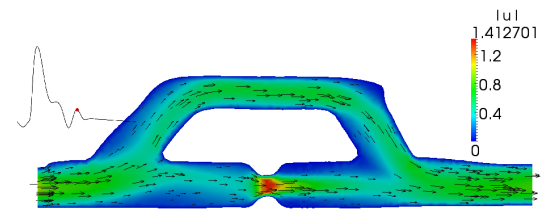
(m) Configurazione ottima, simulazione *c*: modulo della velocità al tempo  $t = t_0$ .



(n) Configurazione ottima, simulazione *c*: modulo della velocità al tempo  $t = t_1$ .

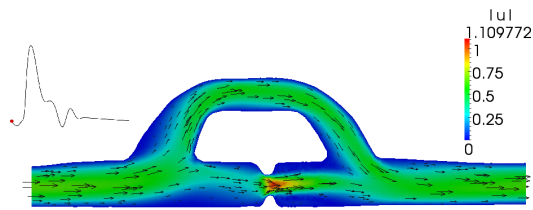


(o) Configurazione ottima, simulazione *c*: modulo della velocità al tempo  $t = t_2$ .

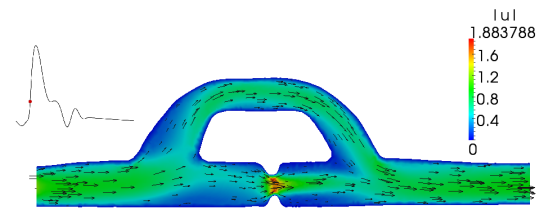


(p) Configurazione ottima, simulazione *c*: modulo della velocità al tempo  $t = t_3$ .

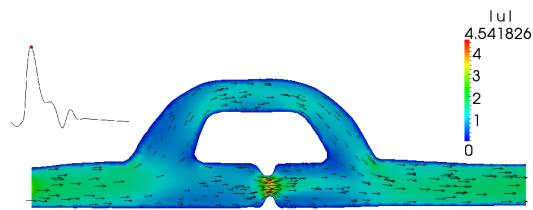
Figura 5.32: Velocità nella sezione longitudinale a diversi istanti temporali (2).



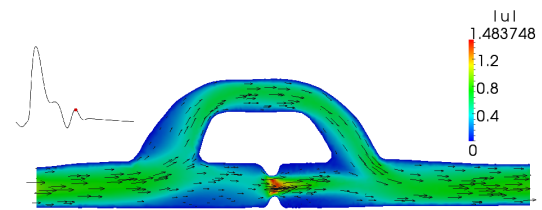
(q) Configurazione ottima, simulazione  $d'$ : modulo della velocità al tempo  $t = t_0$ .



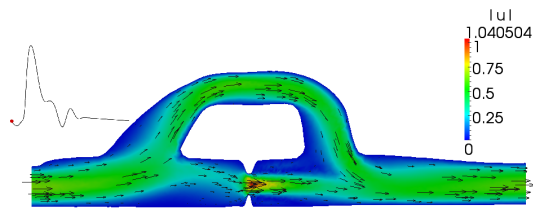
(r) Configurazione ottima, simulazione  $d'$ : modulo della velocità al tempo  $t = t_1$ .



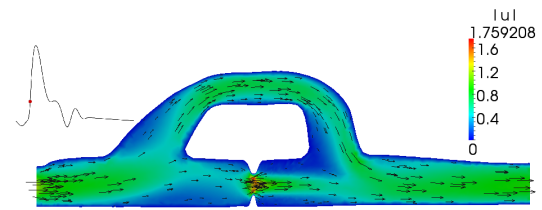
(s) Configurazione ottima, simulazione  $d'$ : modulo della velocità al tempo  $t = t_2$ .



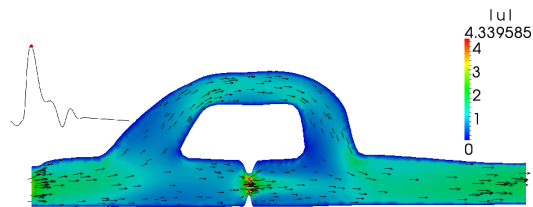
(t) Configurazione ottima, simulazione  $d'$ : modulo della velocità al tempo  $t = t_3$ .



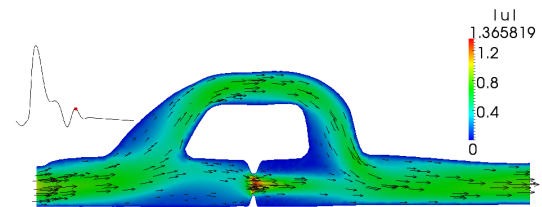
(u) Configurazione ottima, simulazione  $d''$ : modulo della velocità al tempo  $t = t_0$ .



(v) Configurazione ottima, simulazione  $d''$ : modulo della velocità al tempo  $t = t_1$ .

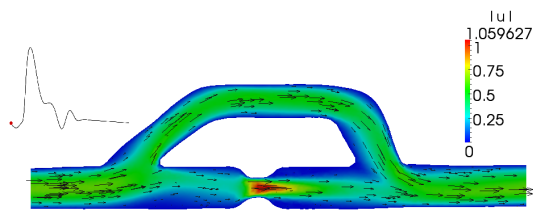


(w) Configurazione ottima, simulazione  $d''$ : modulo della velocità al tempo  $t = t_2$ .

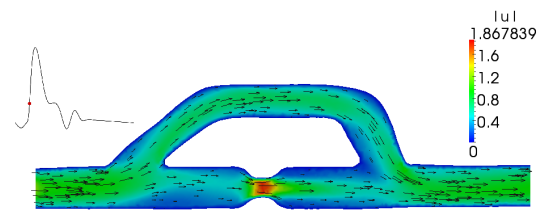


(x) Configurazione ottima, simulazione  $d''$ : modulo della velocità al tempo  $t = t_3$ .

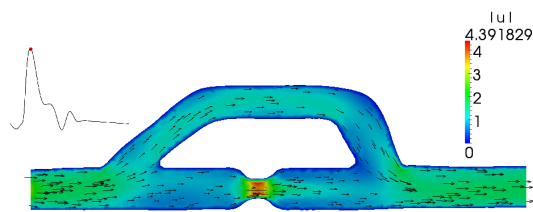
Figura 5.32: Velocità nella sezione longitudinale a diversi istanti temporali (3).



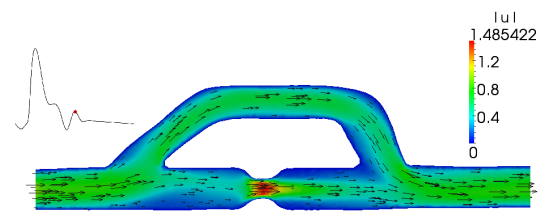
(y) Configurazione ottima, simulazione *e*: modulo della velocità al tempo  $t = t_0$ .



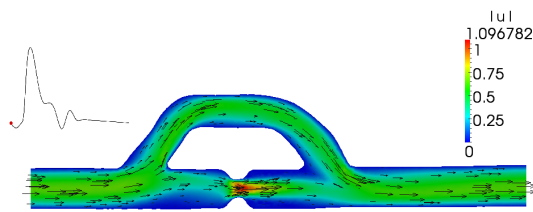
(z) Configurazione ottima, simulazione *e*: modulo della velocità al tempo  $t = t_1$ .



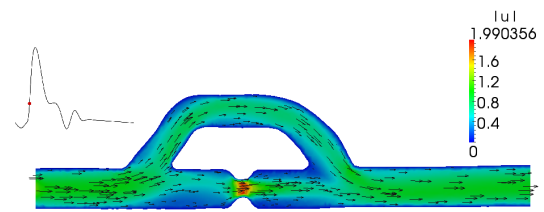
(aa) Configurazione ottima, simulazione *e*: modulo della velocità al tempo  $t = t_2$ .



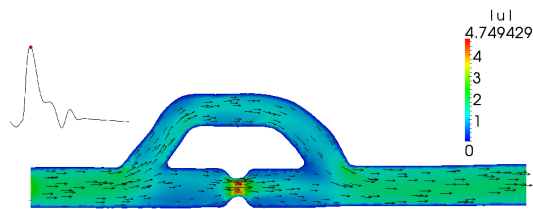
(ab) Configurazione ottima, simulazione *e*: modulo della velocità al tempo  $t = t_3$ .



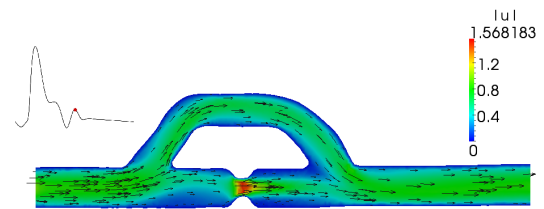
(ac) Configurazione ottima, simulazione *f*: modulo della velocità al tempo  $t = t_0$ .



(ad) Configurazione ottima, simulazione *f*: modulo della velocità al tempo  $t = t_1$ .



(ae) Configurazione ottima, simulazione *f*: modulo della velocità al tempo  $t = t_2$ .



(af) Configurazione ottima, simulazione *f*: modulo della velocità al tempo  $t = t_3$ .

Figura 5.32: Velocità nella sezione longitudinale a diversi istanti temporali (4).

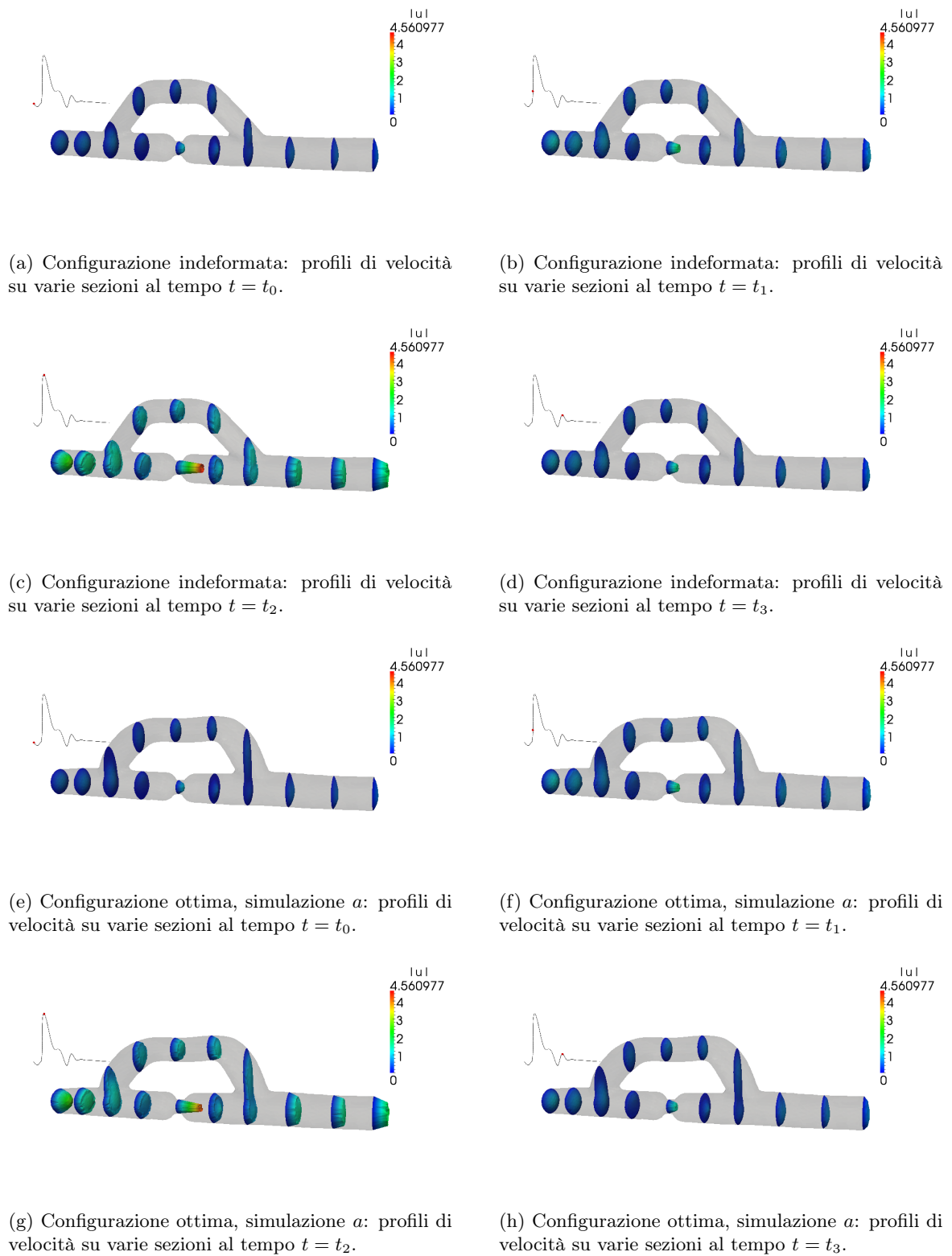
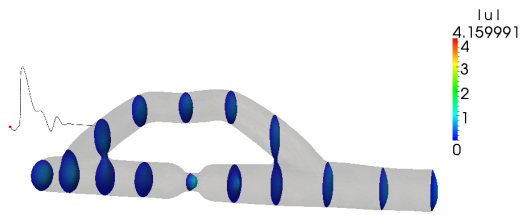
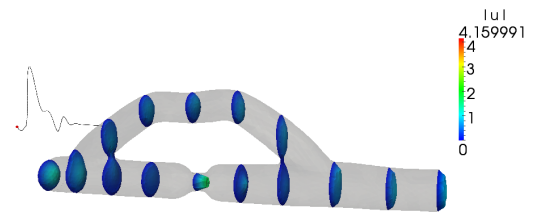


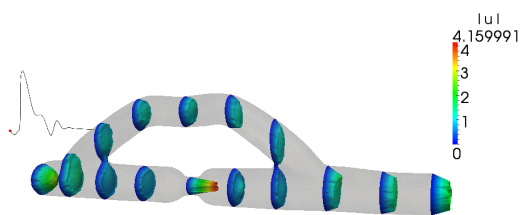
Figura 5.33: Profili di velocità su varie sezioni a diversi istanti temporali.



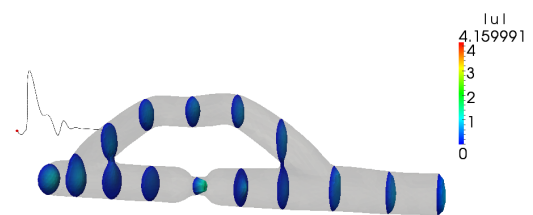
(i) Configurazione ottima, simulazione *b*: profili di velocità su varie sezioni al tempo  $t = t_0$ .



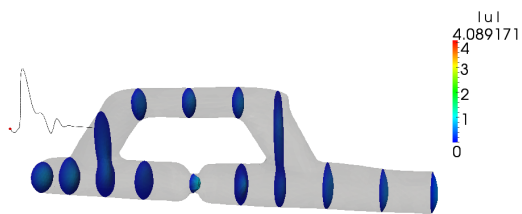
(j) Configurazione ottima, simulazione *b*: profili di velocità su varie sezioni al tempo  $t = t_1$ .



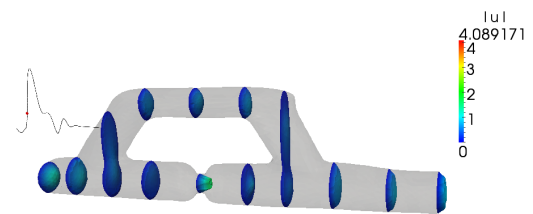
(k) Configurazione ottima, simulazione *b*: profili di velocità su varie sezioni al tempo  $t = t_2$ .



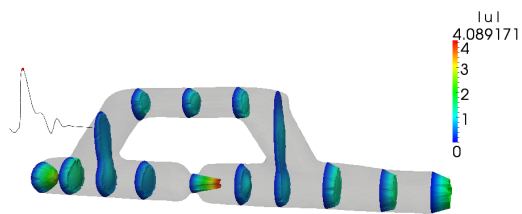
(l) Configurazione ottima, simulazione *b*: profili di velocità su varie sezioni al tempo  $t = t_3$ .



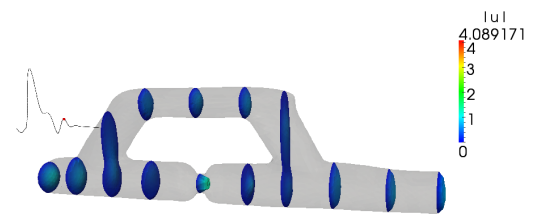
(m) Configurazione ottima, simulazione *c*: profili di velocità su varie sezioni al tempo  $t = t_0$ .



(n) Configurazione ottima, simulazione *c*: profili di velocità su varie sezioni al tempo  $t = t_1$ .

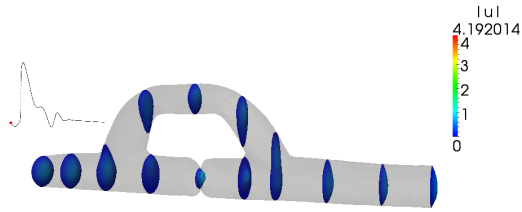


(o) Configurazione ottima, simulazione *c*: profili di velocità su varie sezioni al tempo  $t = t_2$ .

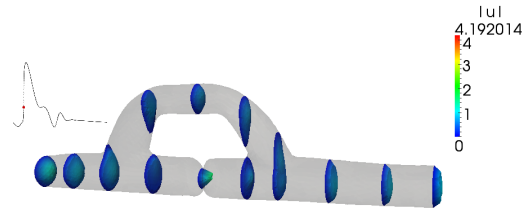


(p) Configurazione ottima, simulazione *c*: profili di velocità su varie sezioni al tempo  $t = t_3$ .

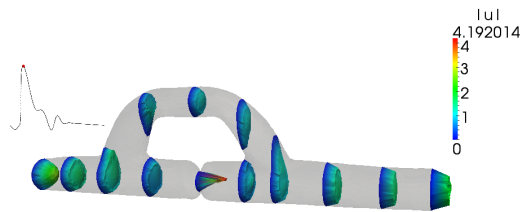
Figura 5.33: Profili di velocità su varie sezioni a diversi istanti temporali (2).



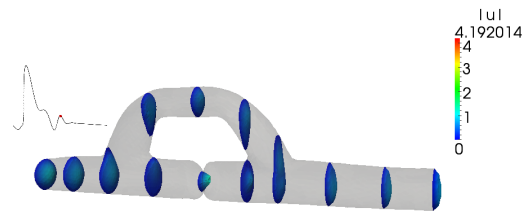
(q) Configurazione ottima, simulazione  $d'$ : profili di velocità su varie sezioni al tempo  $t = t_0$ .



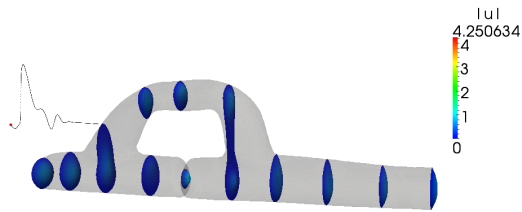
(r) Configurazione ottima, simulazione  $d'$ : profili di velocità su varie sezioni al tempo  $t = t_1$ .



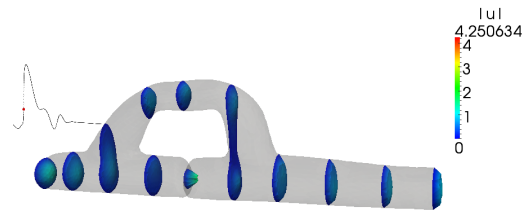
(s) Configurazione ottima, simulazione  $d'$ : profili di velocità su varie sezioni al tempo  $t = t_2$ .



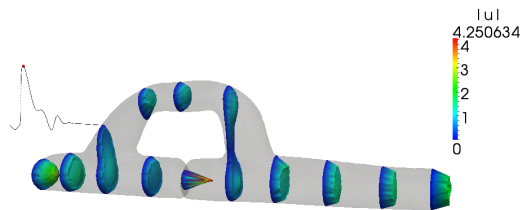
(t) Configurazione ottima, simulazione  $d'$ : profili di velocità su varie sezioni al tempo  $t = t_3$ .



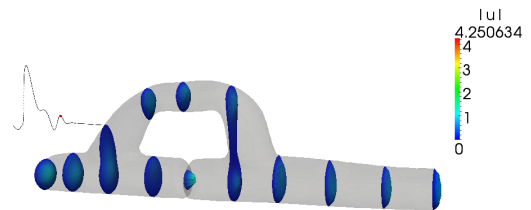
(u) Configurazione ottima, simulazione  $d''$ : profili di velocità su varie sezioni al tempo  $t = t_0$ .



(v) Configurazione ottima, simulazione  $d''$ : profili di velocità su varie sezioni al tempo  $t = t_1$ .



(w) Configurazione ottima, simulazione  $d''$ : profili di velocità su varie sezioni al tempo  $t = t_2$ .



(x) Configurazione ottima, simulazione  $d''$ : profili di velocità su varie sezioni al tempo  $t = t_3$ .

Figura 5.33: Profili di velocità su varie sezioni a diversi istanti temporali (3).



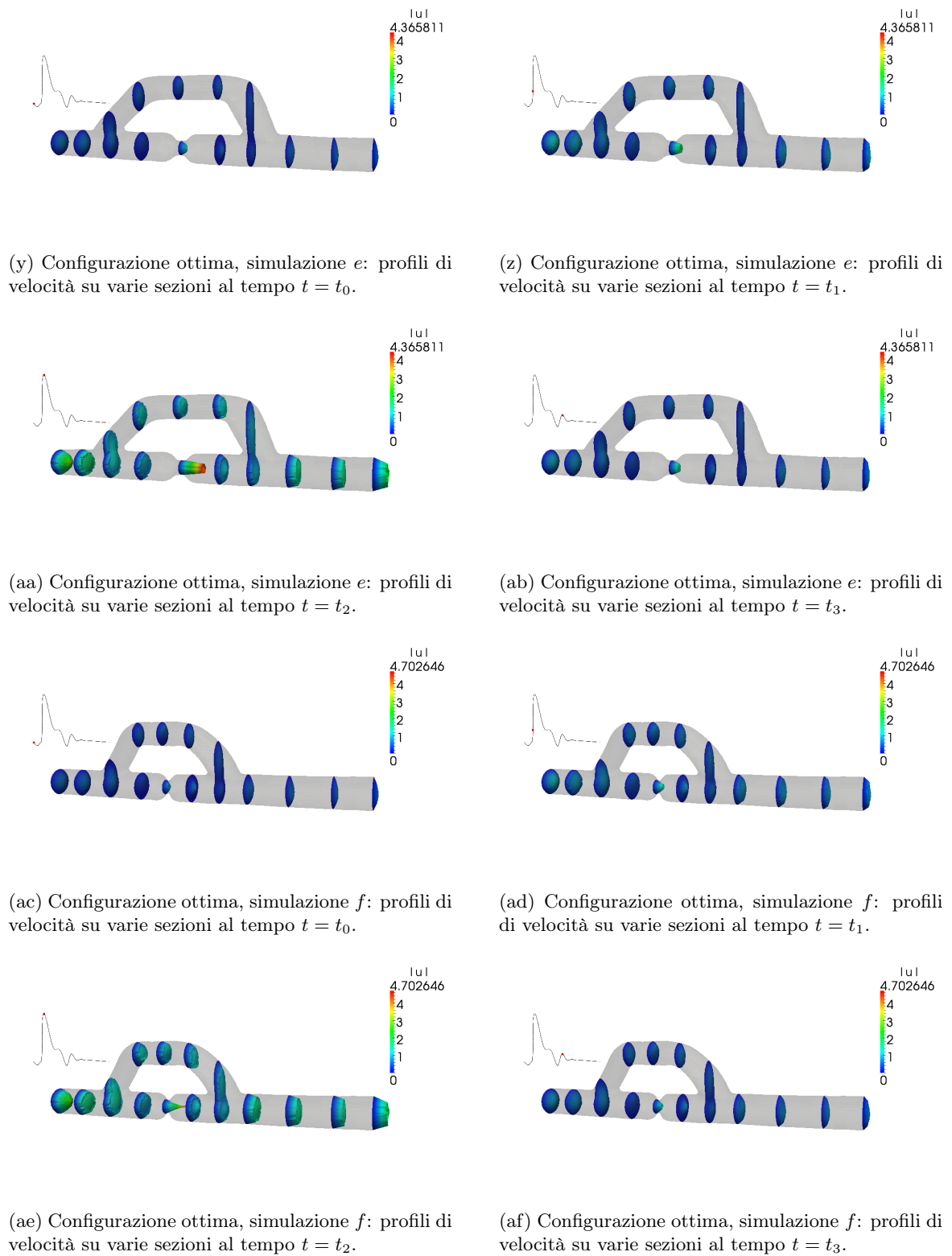
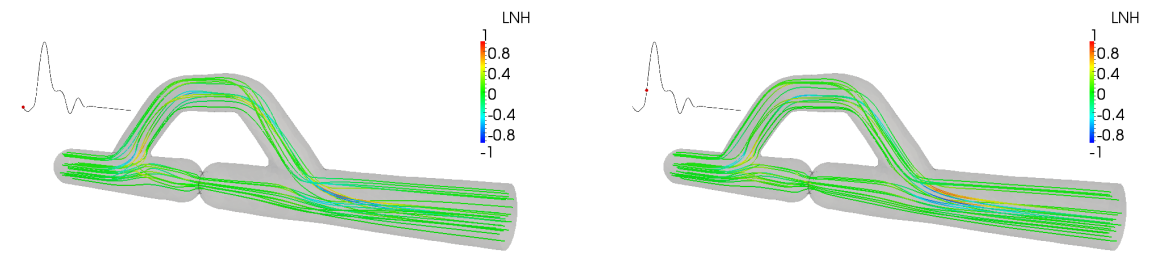
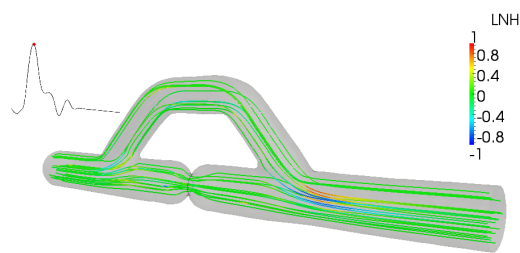


Figura 5.33: Profili di velocità su varie sezioni a diversi istanti temporali (4).

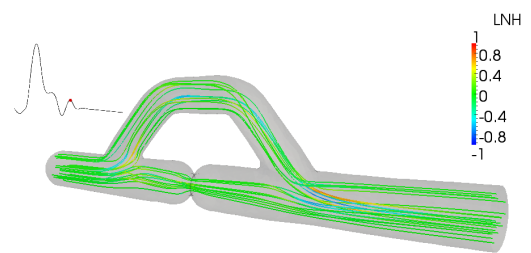


(a) Configurazione indeformata: *streamlines* al tempo  $t = t_0$ .

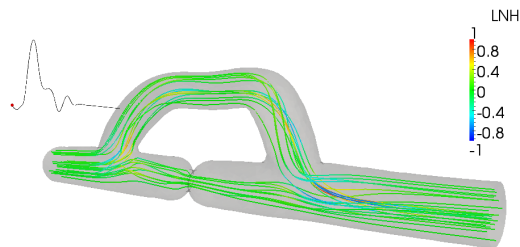
(b) Configurazione indeformata: *streamlines* al tempo  $t = t_1$ .



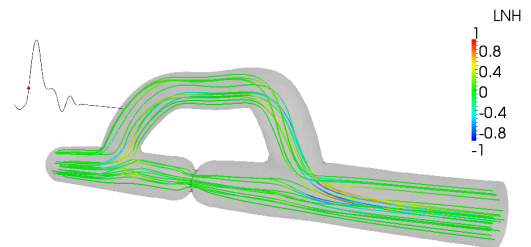
(c) Configurazione indeformata: *streamlines* al tempo  $t = t_2$ .



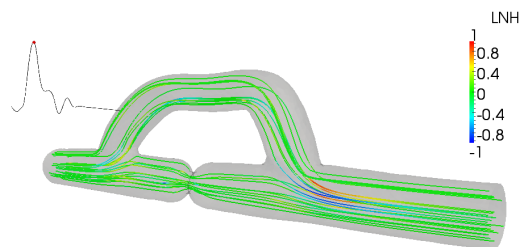
(d) Configurazione indeformata: *streamlines* al tempo  $t = t_3$ .



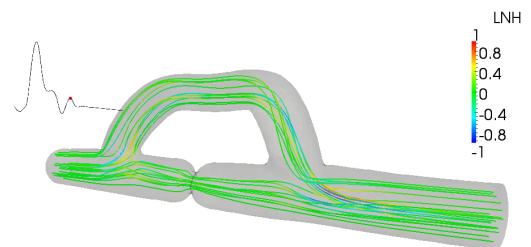
(e) Configurazione ottima, simulazione *a*: *streamlines* al tempo  $t = t_0$ .



(f) Configurazione ottima, simulazione *a*: *streamlines* al tempo  $t = t_1$ .

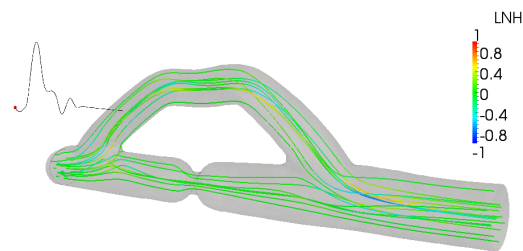


(g) Configurazione ottima, simulazione *a*: *streamlines* al tempo  $t = t_2$ .

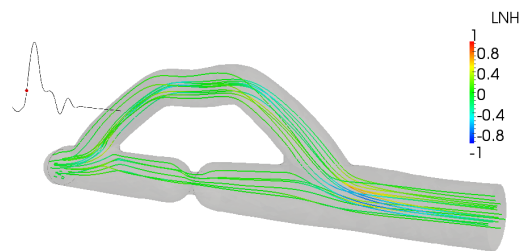


(h) Configurazione ottima, simulazione *a*: *streamlines* al tempo  $t = t_3$ .

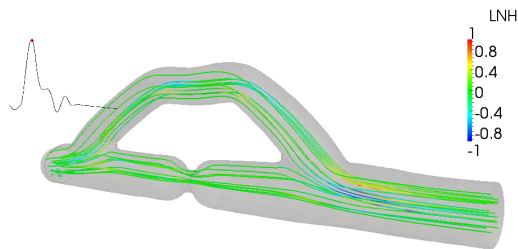
Figura 5.34: *Streamlines* a diversi istanti temporali.



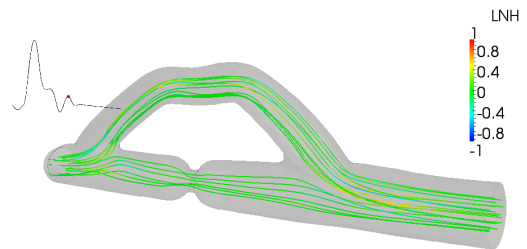
(i) Configurazione ottima, simulazione *b*: *streamlines* al tempo  $t = t_0$ .



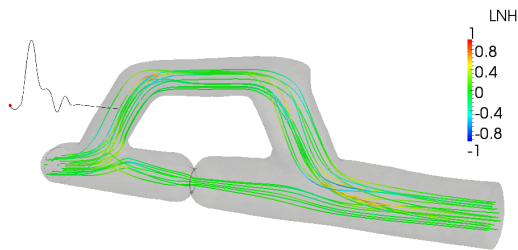
(j) Configurazione ottima, simulazione *b*: *streamlines* al tempo  $t = t_1$ .



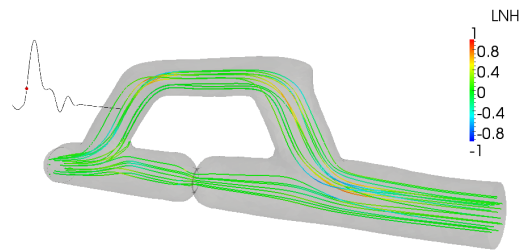
(k) Configurazione ottima, simulazione *b*: *streamlines* al tempo  $t = t_2$ .



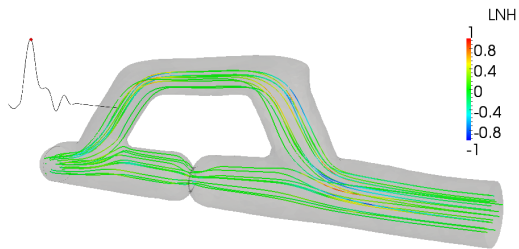
(l) Configurazione ottima, simulazione *b*: *streamlines* al tempo  $t = t_3$ .



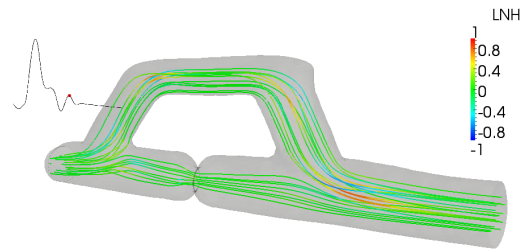
(m) Configurazione ottima, simulazione *c*: *streamlines* al tempo  $t = t_0$ .



(n) Configurazione ottima, simulazione *c*: *streamlines* al tempo  $t = t_1$ .

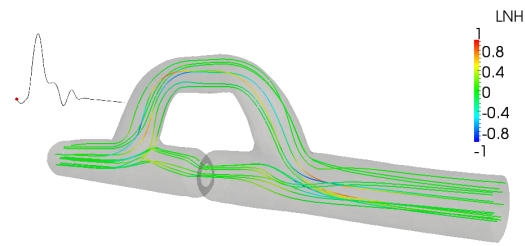


(o) Configurazione ottima, simulazione *c*: *streamlines* al tempo  $t = t_2$ .

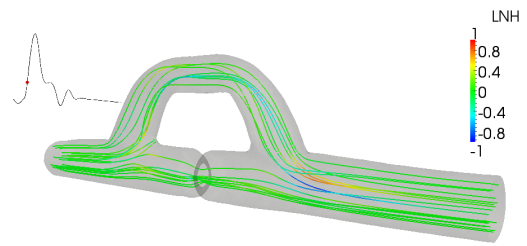


(p) Configurazione ottima, simulazione *c*: *streamlines* al tempo  $t = t_3$ .

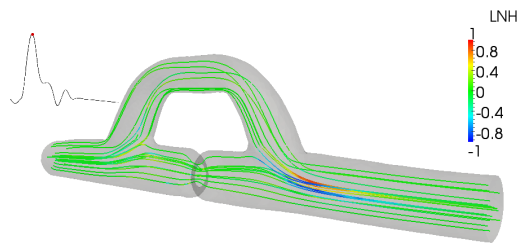
Figura 5.34: *Streamlines* a diversi istanti temporali (2).



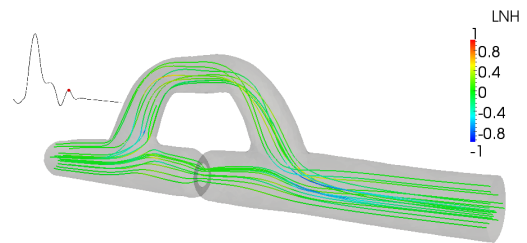
(q) Configurazione ottima, simulazione  $d'$ : *streamlines* al tempo  $t = t_0$ .



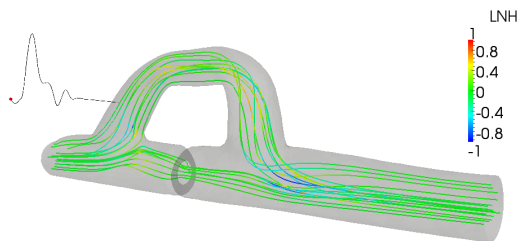
(r) Configurazione ottima, simulazione  $d'$ : *streamlines* al tempo  $t = t_1$ .



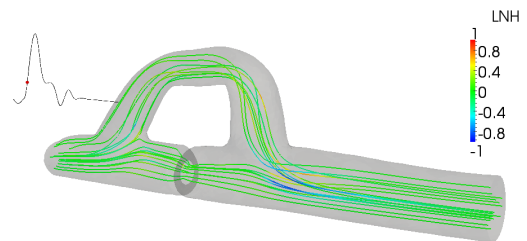
(s) Configurazione ottima, simulazione  $d'$ : *streamlines* al tempo  $t = t_2$ .



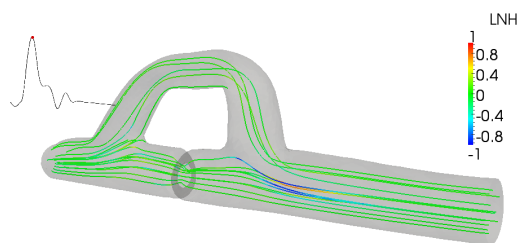
(t) Configurazione ottima, simulazione  $d'$ : *streamlines* al tempo  $t = t_3$ .



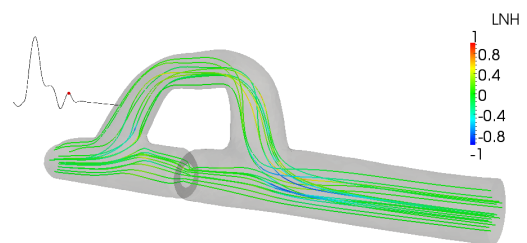
(u) Configurazione ottima, simulazione  $d''$ : *streamlines* al tempo  $t = t_0$ .



(v) Configurazione ottima, simulazione  $d''$ : *streamlines* al tempo  $t = t_1$ .

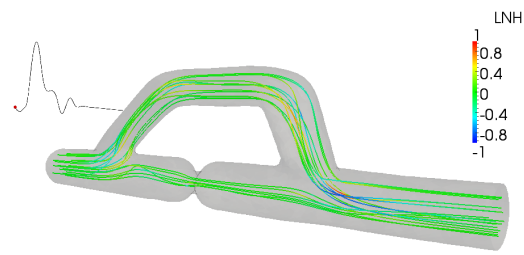


(w) Configurazione ottima, simulazione  $d''$ : *streamlines* al tempo  $t = t_2$ .

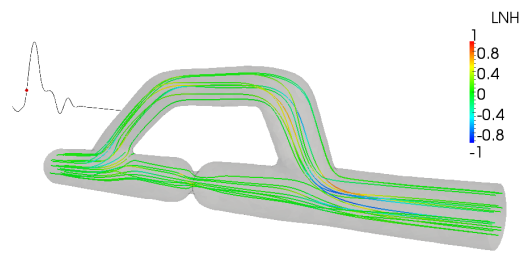


(x) Configurazione ottima, simulazione  $d''$ : *streamlines* al tempo  $t = t_3$ .

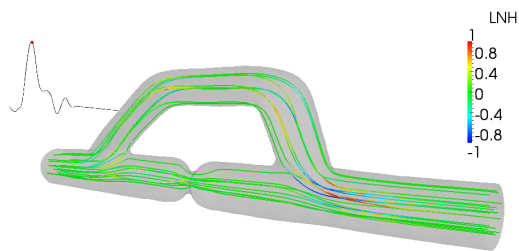
Figura 5.34: *Streamlines* a diversi istanti temporali (3).



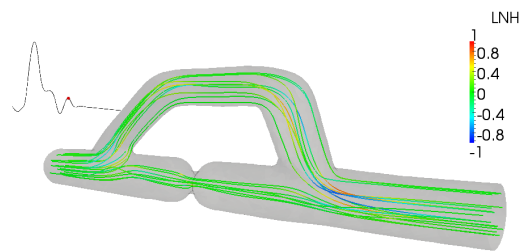
(y) Configurazione ottima, simulazione *e*: *streamlines* al tempo  $t = t_0$ .



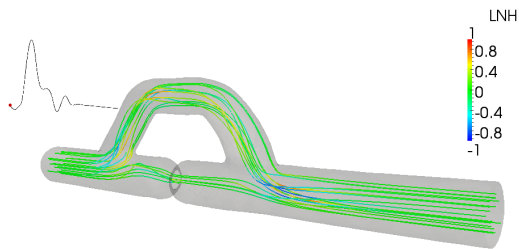
(z) Configurazione ottima, simulazione *e*: *streamlines* al tempo  $t = t_1$ .



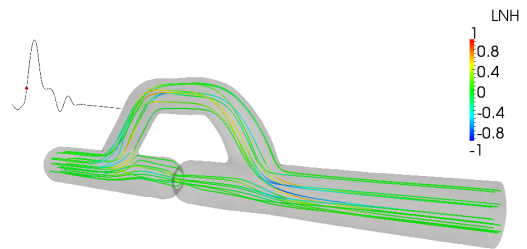
(aa) Configurazione ottima, simulazione *e*: *streamlines* al tempo  $t = t_2$ .



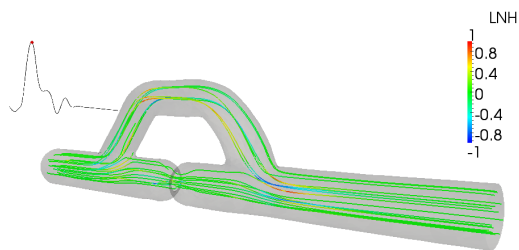
(ab) Configurazione ottima, simulazione *e*: *streamlines* al tempo  $t = t_3$ .



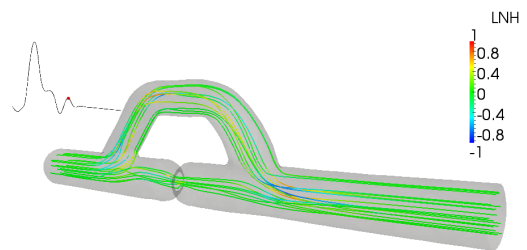
(ac) Configurazione ottima, simulazione *f*: *streamlines* al tempo  $t = t_0$ .



(ad) Configurazione ottima, simulazione *f*: *streamlines* al tempo  $t = t_1$ .

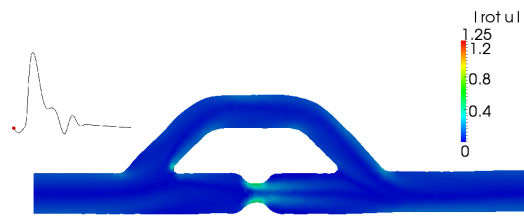


(ae) Configurazione ottima, simulazione *f*: *streamlines* al tempo  $t = t_2$ .

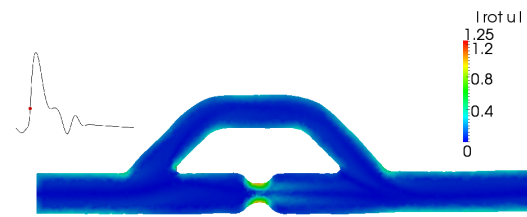


(af) Configurazione ottima, simulazione *f*: *streamlines* al tempo  $t = t_3$ .

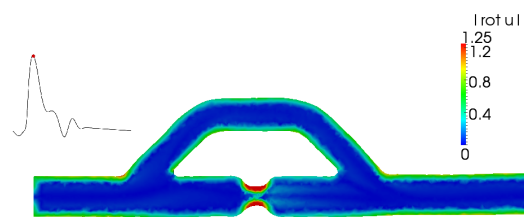
Figura 5.34: *Streamlines* a diversi istanti temporali (4).



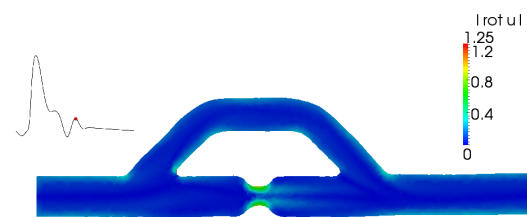
(a) Configurazione indeformata: modulo della vorticità al tempo  $t = t_0$ .



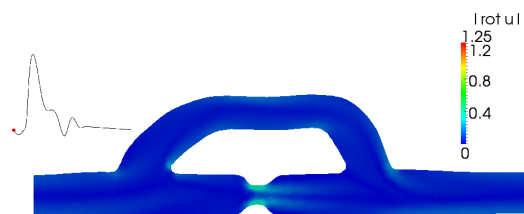
(b) Configurazione indeformata: modulo della vorticità al tempo  $t = t_1$ .



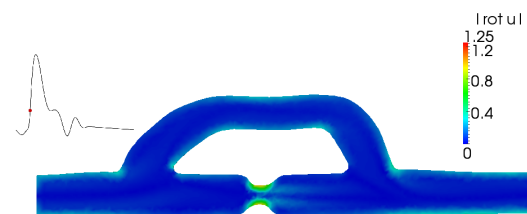
(c) Configurazione indeformata: modulo della vorticità al tempo  $t = t_2$ .



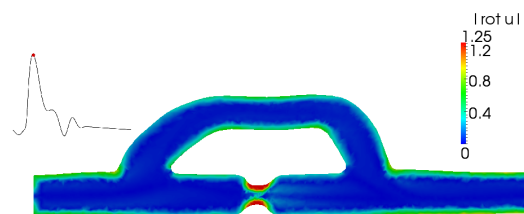
(d) Configurazione indeformata: modulo della vorticità al tempo  $t = t_3$ .



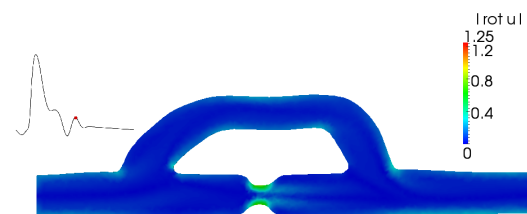
(e) Configurazione ottima, simulazione *a*: modulo della vorticità al tempo  $t = t_0$ .



(f) Configurazione ottima, simulazione *a*: modulo della vorticità al tempo  $t = t_1$ .

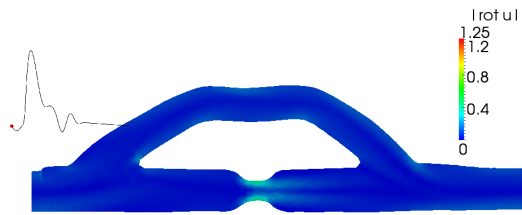


(g) Configurazione ottima, simulazione *a*: modulo della vorticità al tempo  $t = t_2$ .

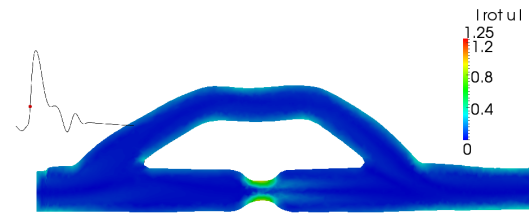


(h) Configurazione ottima, simulazione *a*: modulo della vorticità al tempo  $t = t_3$ .

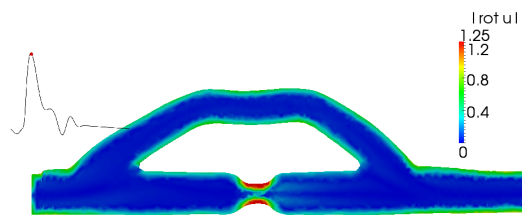
Figura 5.35: Modulo della vorticità a diversi istanti temporali.



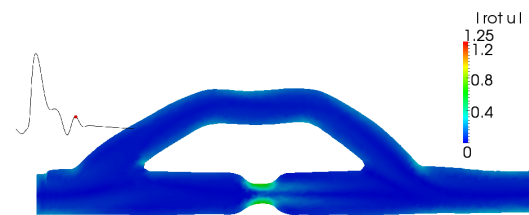
(i) Configurazione ottima, simulazione *b*: modulo della vorticità al tempo  $t = t_0$ .



(j) Configurazione ottima, simulazione *b*: modulo della vorticità al tempo  $t = t_1$ .



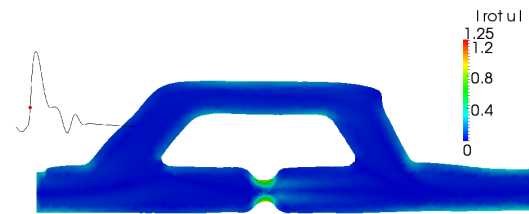
(k) Configurazione ottima, simulazione *b*: modulo della vorticità al tempo  $t = t_2$ .



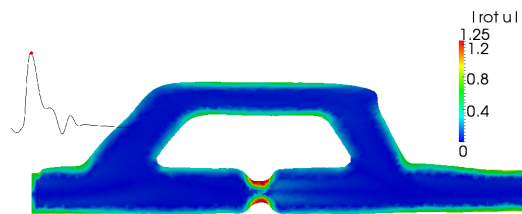
(l) Configurazione ottima, simulazione *b*: modulo della vorticità al tempo  $t = t_3$ .



(m) Configurazione ottima, simulazione *c*: modulo della vorticità al tempo  $t = t_0$ .



(n) Configurazione ottima, simulazione *c*: modulo della vorticità al tempo  $t = t_1$ .



(o) Configurazione ottima, simulazione *c*: modulo della vorticità al tempo  $t = t_2$ .



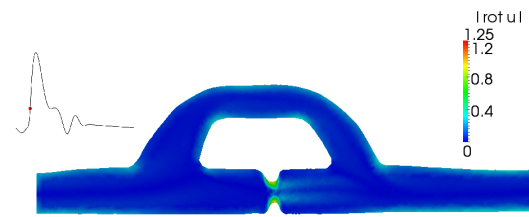
(p) Configurazione ottima, simulazione *c*: modulo della vorticità al tempo  $t = t_3$ .

Figura 5.35: Modulo della vorticità a diversi istanti temporali (2).

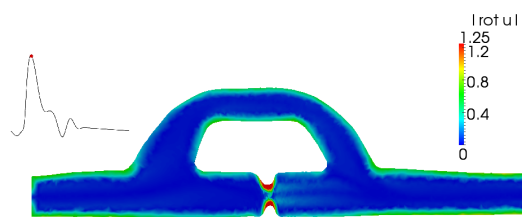




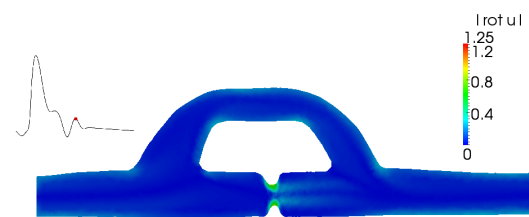
(q) Configurazione ottima, simulazione  $d'$ : modulo della vorticità al tempo  $t = t_0$ .



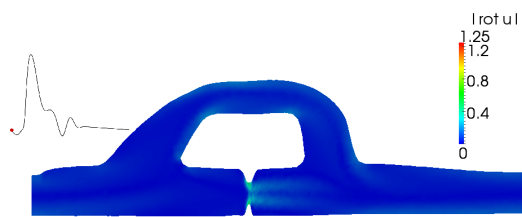
(r) Configurazione ottima, simulazione  $d'$ : modulo della vorticità al tempo  $t = t_1$ .



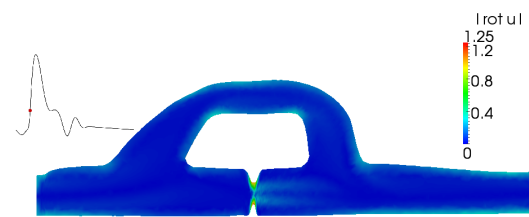
(s) Configurazione ottima, simulazione  $d'$ : modulo della vorticità al tempo  $t = t_2$ .



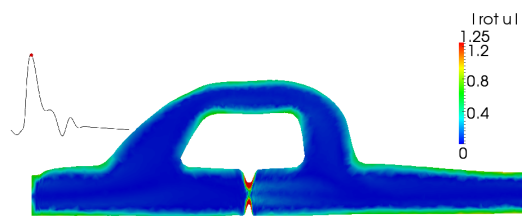
(t) Configurazione ottima, simulazione  $d'e$ : modulo della vorticità al tempo  $t = t_3$ .



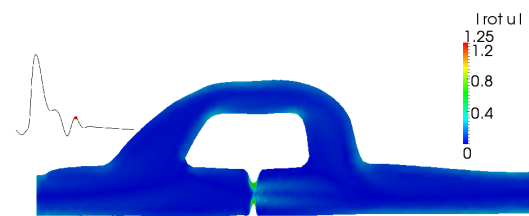
(u) Configurazione ottima, simulazione  $d''$ : modulo della vorticità al tempo  $t = t_0$ .



(v) Configurazione ottima, simulazione  $d''$ : modulo della vorticità al tempo  $t = t_1$ .



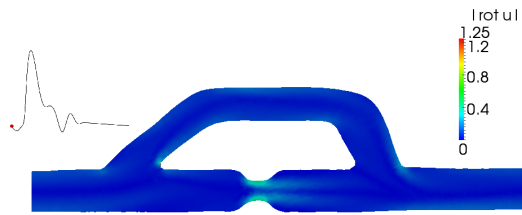
(w) Configurazione ottima, simulazione  $d''$ : modulo della vorticità al tempo  $t = t_2$ .



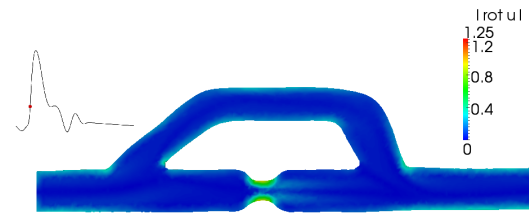
(x) Configurazione ottima, simulazione  $d''$ : modulo della vorticità al tempo  $t = t_3$ .

Figura 5.35: Modulo della vorticità a diversi istanti temporali (3).

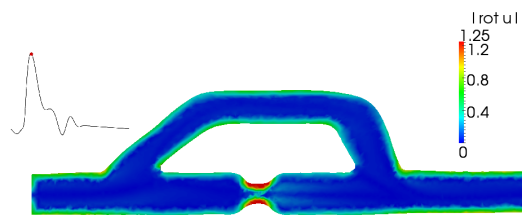




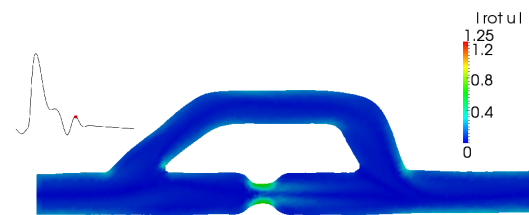
(y) Configurazione ottima, simulazione *e*: modulo della vorticità al tempo  $t = t_0$ .



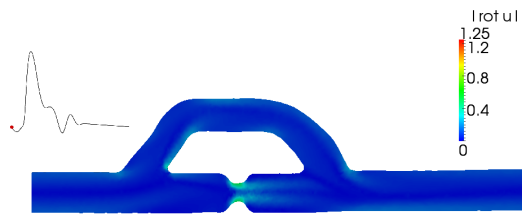
(z) Configurazione ottima, simulazione *e*: modulo della vorticità al tempo  $t = t_1$ .



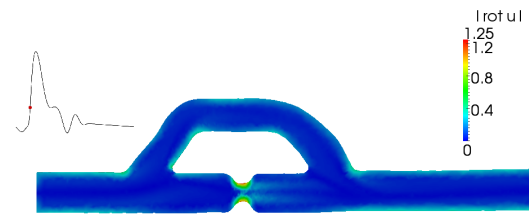
(aa) Configurazione ottima, simulazione *e*: modulo della vorticità al tempo  $t = t_2$ .



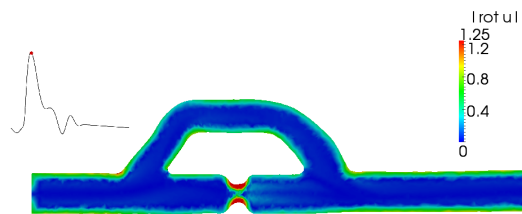
(ab) Configurazione ottima, simulazione *e*: modulo della vorticità al tempo  $t = t_3$ .



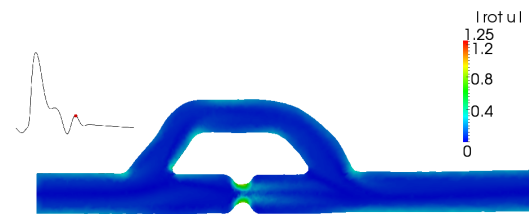
(ac) Configurazione ottima, simulazione *f*: modulo della vorticità al tempo  $t = t_0$ .



(ad) Configurazione ottima, simulazione *f*: modulo della vorticità al tempo  $t = t_1$ .

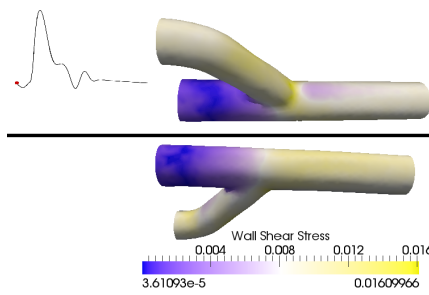


(ae) Configurazione ottima, simulazione *f*: modulo della vorticità al tempo  $t = t_2$ .

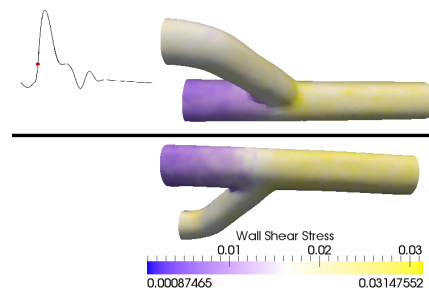


(af) Configurazione ottima, simulazione *f*: modulo della vorticità al tempo  $t = t_3$ .

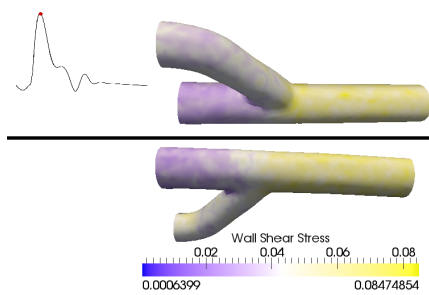
Figura 5.35: Modulo della vorticità a diversi istanti temporali (4).



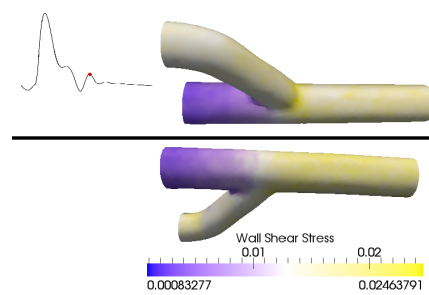
(a) Configurazione indeformata: modulo della sforzo di taglio a parete al tempo  $t = t_0$ .



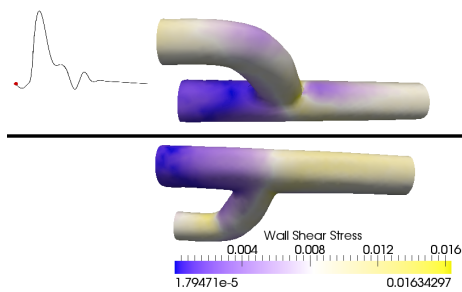
(b) Configurazione indeformata: modulo della sforzo di taglio a parete al tempo  $t = t_1$ .



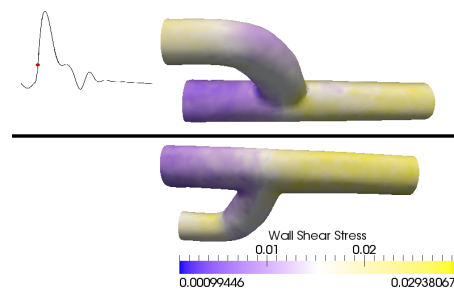
(c) Configurazione indeformata: modulo della sforzo di taglio a parete al tempo  $t = t_2$ .



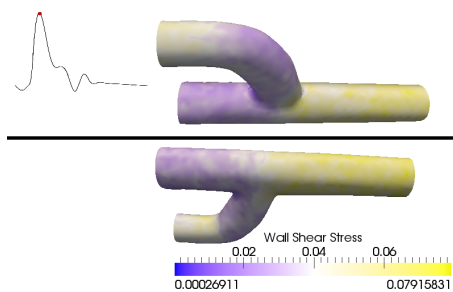
(d) Configurazione indeformata: modulo della sforzo di taglio a parete al tempo  $t = t_3$ .



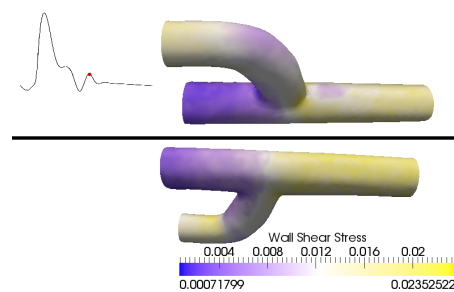
(e) Configurazione ottima, simulazione *a*: modulo della sforzo di taglio a parete al tempo  $t = t_0$ .



(f) Configurazione ottima, simulazione *a*: modulo della sforzo di taglio a parete al tempo  $t = t_1$ .

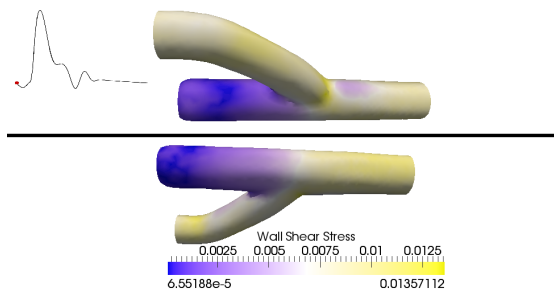


(g) Configurazione ottima, simulazione *a*: modulo della sforzo di taglio a parete al tempo  $t = t_2$ .

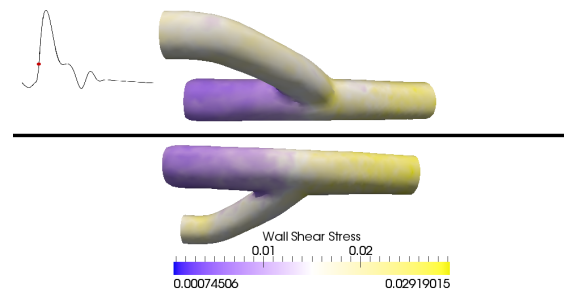


(h) Configurazione ottima, simulazione *a*: modulo della sforzo di taglio a parete al tempo  $t = t_3$ .

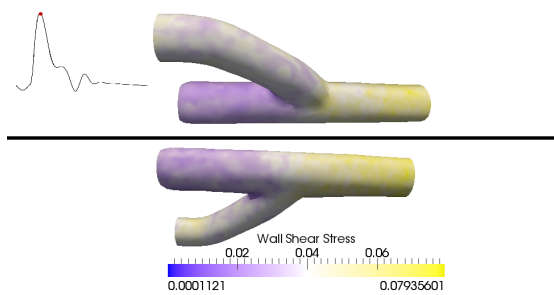
Figura 5.36: Modulo della sforzo di taglio a parete nella regione a valle della stenosi, a diversi istanti temporali.



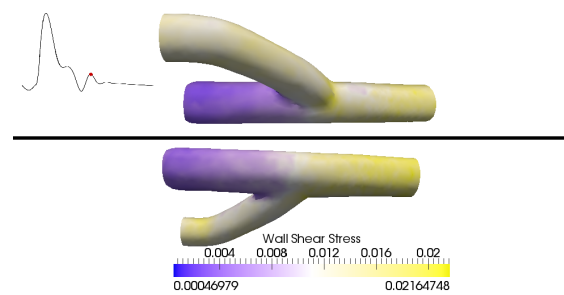
(i) Configurazione ottima, simulazione *b*: modulo della sforzo di taglio a parete al tempo  $t = t_0$ .



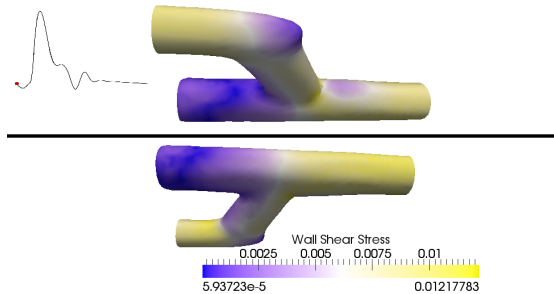
(j) Configurazione ottima, simulazione *b*: modulo della sforzo di taglio a parete al tempo  $t = t_1$ .



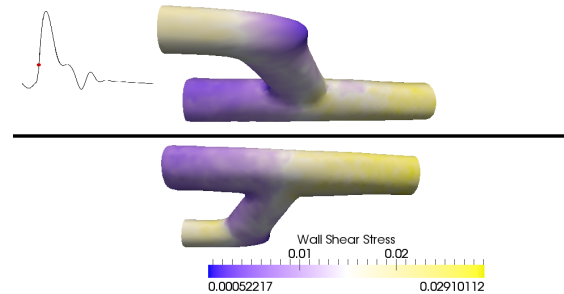
(k) Configurazione ottima, simulazione *b*: modulo della sforzo di taglio a parete al tempo  $t = t_2$ .



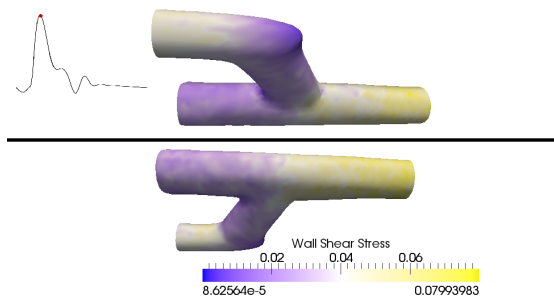
(l) Configurazione ottima, simulazione *b*: modulo della sforzo di taglio a parete al tempo  $t = t_3$ .



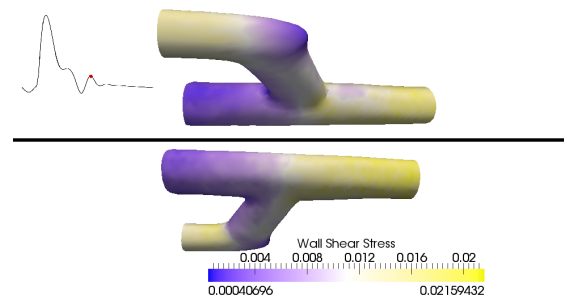
(m) Configurazione ottima, simulazione *c*: modulo della sforzo di taglio a parete al tempo  $t = t_0$ .



(n) Configurazione ottima, simulazione *c*: modulo della sforzo di taglio a parete al tempo  $t = t_1$ .

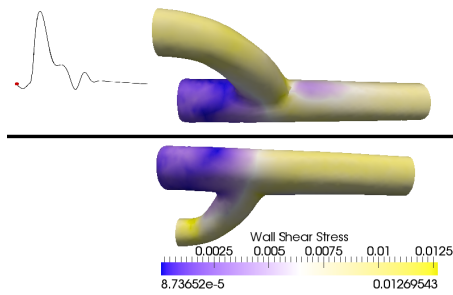


(o) Configurazione ottima, simulazione *c*: modulo della sforzo di taglio a parete al tempo  $t = t_2$ .

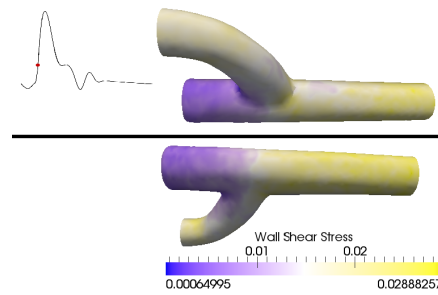


(p) Configurazione ottima, simulazione *c*: modulo della sforzo di taglio a parete al tempo  $t = t_3$ .

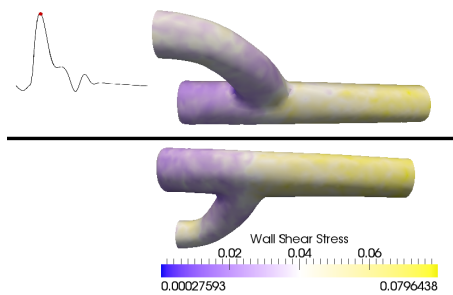
Figura 5.36: Modulo della sforzo di taglio a parete nella regione a valle della stenosi, a diversi istanti temporali (2).



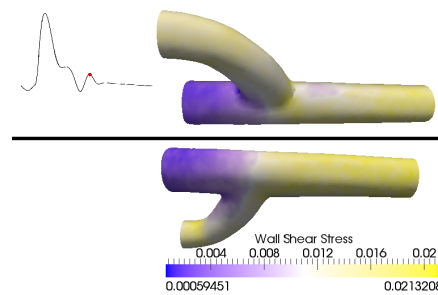
(q) Configurazione ottima, simulazione  $d'$ : modulo della sforzo di taglio a parete al tempo  $t = t_0$ .



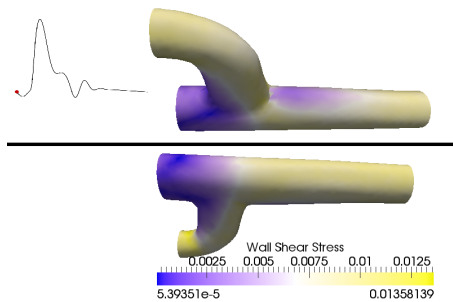
(r) Configurazione ottima, simulazione  $d'$ : modulo della sforzo di taglio a parete al tempo  $t = t_1$ .



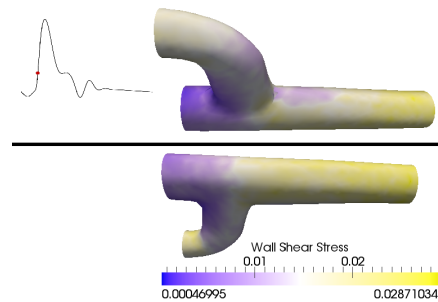
(s) Configurazione ottima, simulazione  $d'$ : modulo della sforzo di taglio a parete al tempo  $t = t_2$ .



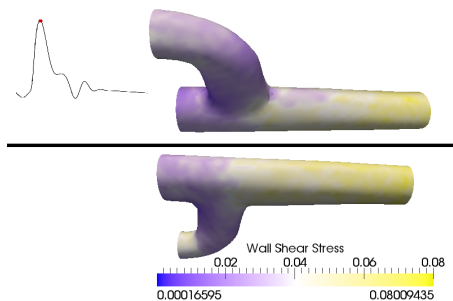
(t) Configurazione ottima, simulazione  $d'$ : modulo della sforzo di taglio a parete al tempo  $t = t_3$ .



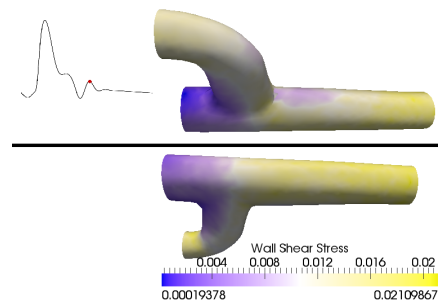
(u) Configurazione ottima, simulazione  $d''$ : modulo della sforzo di taglio a parete al tempo  $t = t_0$ .



(v) Configurazione ottima, simulazione  $d''$ : modulo della sforzo di taglio a parete al tempo  $t = t_1$ .

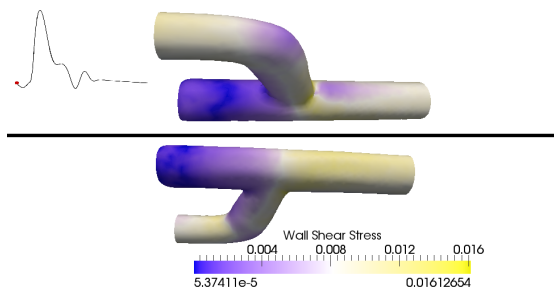


(w) Configurazione ottima, simulazione  $d''$ : modulo della sforzo di taglio a parete al tempo  $t = t_2$ .

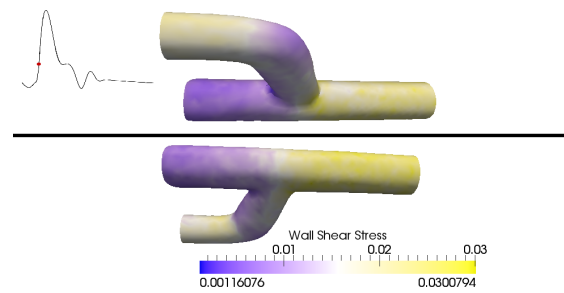


(x) Configurazione ottima, simulazione  $d''$ : modulo della sforzo di taglio a parete al tempo  $t = t_3$ .

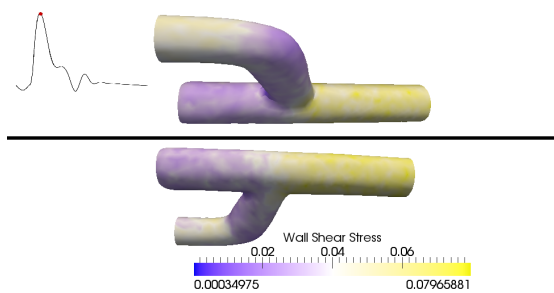
Figura 5.36: Modulo della sforzo di taglio a parete nella regione a valle della stenosi, a diversi istanti temporali (3).



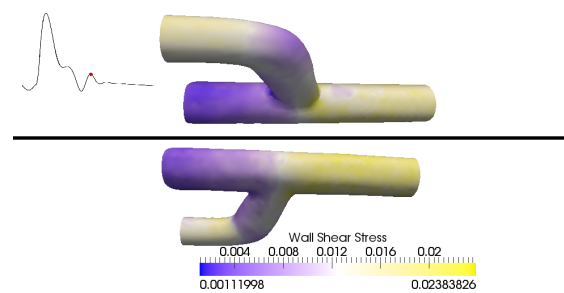
(y) Configurazione ottima, simulazione *e*: modulo della sforzo di taglio a parete al tempo  $t = t_0$ .



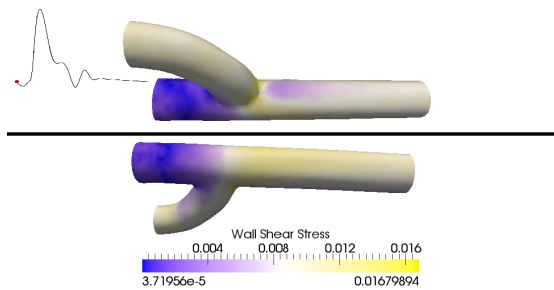
(z) Configurazione ottima, simulazione *e*: modulo della sforzo di taglio a parete al tempo  $t = t_1$ .



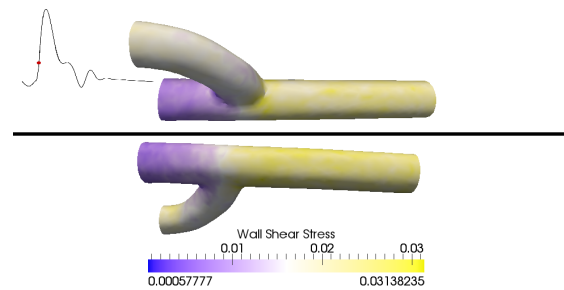
(aa) Configurazione ottima, simulazione *e*: modulo della sforzo di taglio a parete al tempo  $t = t_2$ .



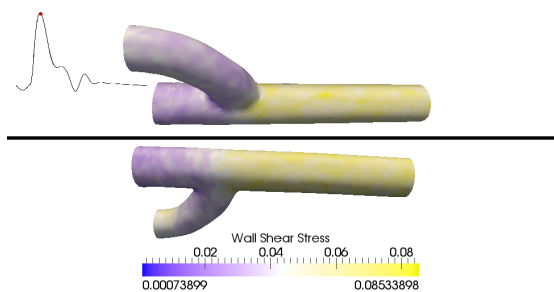
(ab) Configurazione ottima, simulazione *e*: modulo della sforzo di taglio a parete al tempo  $t = t_3$ .



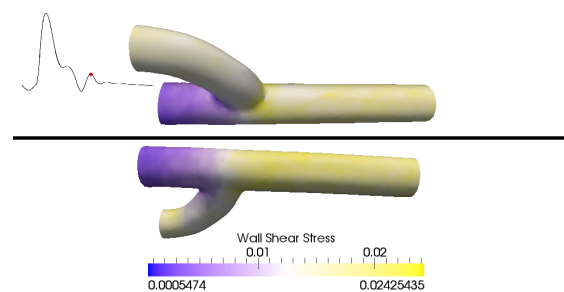
(ac) Configurazione ottima, simulazione *f*: modulo della sforzo di taglio a parete al tempo  $t = t_0$ .



(ad) Configurazione ottima, simulazione *f*: modulo della sforzo di taglio a parete al tempo  $t = t_1$ .

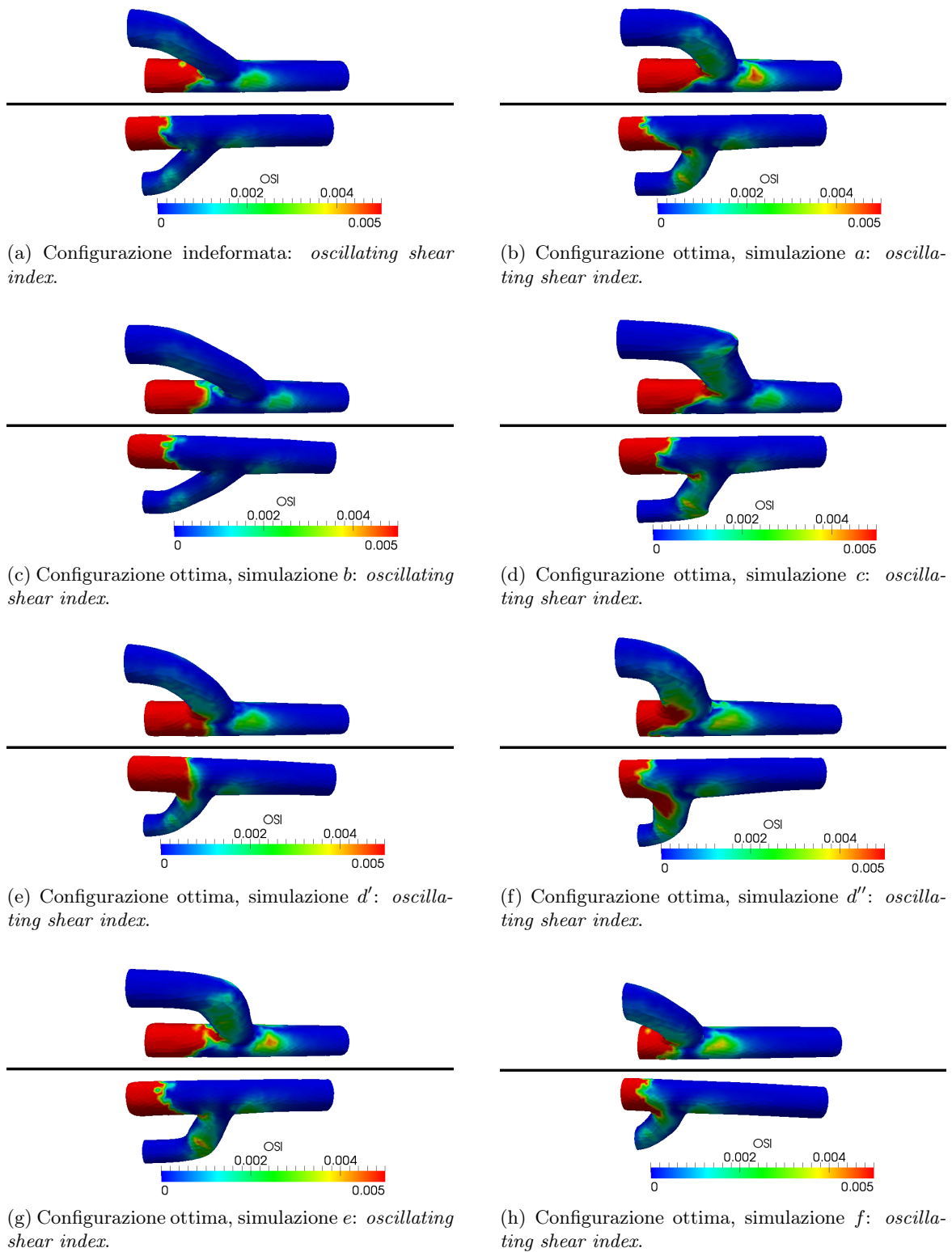


(ae) Configurazione ottima, simulazione *f*: modulo della sforzo di taglio a parete al tempo  $t = t_2$ .



(af) Configurazione ottima, simulazione *f*: modulo della sforzo di taglio a parete al tempo  $t = t_3$ .

Figura 5.36: Modulo della sforzo di taglio a parete nella regione a valle della stenosi, a diversi istanti temporali (4).

Figura 5.37: *Oscillating shear index*.

# Conclusioni

In questa Tesi di Laurea Magistrale sono stati studiati alcuni metodi analitici e numerici per la risoluzione di problemi di ottimizzazione di forma.

La prima fase di questo lavoro, presentata nel CAPITOLO 2, ha riguardato lo studio degli strumenti per l'analisi di un problema di ottimizzazione, visto come caso particolare di problema di controllo ottimo per EDP. Dopo averne descritto una formulazione astratta, è stato enunciato un risultato, fondamentale, di esistenza, di cui sono state discusse le principali ipotesi (una proprietà di compattezza della topologia introdotta sull'insieme di forme ammissibili, la continuità della soluzione del modello fisico rispetto alle variazioni del dominio e la semicontinuità inferiore del funzionale costo). Sono stati poi presentati due differenti approcci per la caratterizzazione delle condizioni di ottimalità: un primo metodo basato sull'*approccio alla Lions* ed un secondo approccio basato sul metodo dei *moltiplicatori di Lagrange*. Quest'ultimo metodo risulta particolarmente vantaggioso nei problemi di ottimizzazione di forma, perché permette di esprimere facilmente il gradiente del funzionale costo senza ricorrere al calcolo della *shape derivative* del problema di stato.

Gli strumenti teorici introdotti si sono rivelati molto potenti e versatili: la loro applicazione, in particolare, a problemi di ottimizzazione di forma in un ambito fluidodinamico è stata descritta nel CAPITOLO 3. Il caso discusso in dettaglio ha riguardato la minimizzazione dell'energia dissipata per un fluido viscoso incomprimibile governato dalle equazioni di Stokes (stazionarie e non stazionarie). Dopo aver dimostrato l'esistenza di (almeno) una forma ottima, mediante applicazione del risultato astratto di esistenza, sono state derivate le condizioni di ottimalità, sfruttate successivamente per la risoluzione numerica del problema.

La seconda parte di questo lavoro, presentata nel CAPITOLO 4, ha riguardato lo studio di metodi iterativi per l'approssimazione numerica dei problemi di ottimizzazione di forma, con particolare attenzione alla riduzione dimensionale nel trattamento delle forme ammissibili e delle relative deformazioni. La parametrizzazione basata su mappe di tipo *Free Form Deformation* ha permesso di descrivere in modo computazionalmente efficiente un'ampia classe di forme ammissibili mediante un ristretto numero di parametri. Inoltre, questa tecnica non richiede alcuna operazione di *remeshing* ed esegue deformazioni che preservano le proprietà di regolarità richieste. Infine, è stato descritto come ricavare, a partire dalle informazioni contenute nello *shape gradient*, le condizioni di ottimalità direttamente rispetto ai parametri della mappa.

La terza parte del lavoro di Tesi ha contribuito all'implementazione, a partire dalla libreria *LifeV*, di un risolutore in C++ per un generico problema di ottimizzazione di forma con parametrizzazione *Free Form Deformation*, presentato in dettaglio nell'APPENDICE A. Grazie a questo codice, è possibile risolvere una ampia classe di problemi di ottimizzazione di forma: la versatilità della mappa *Free Form Deformation* permette di considerare un vasto insieme di possibili deformazioni della geometria, le classi fornite da *LifeV* permettono di considerare numerosi modelli differenti per il problema di stato e la struttura della libreria garantisce una facile estendibilità

per quanto riguarda il metodo di risoluzione del problema di ottimizzazione rispetto ai parametri della mappa *Free Form Deformation*.

Questo risolutore è stato applicato ad alcuni problemi di ottimizzazione di forma, come descritto nel CAPITOLO 5: è stato anzitutto presentato un caso test, per il quale è disponibile analiticamente l'espressione di una forma ottima, da un lato per verificare la correttezza dell'implementazione, e dall'altro per studiare l'influenza della scelta dei gradi di libertà della mappa *Free Form Deformation*. Successivamente, è stata descritta l'applicazione della libreria alla risoluzione di problemi di ottimizzazione di forma in fluidodinamica, partendo dal caso, molto studiato in letteratura, di minimizzazione dell'energia dissipata (o, in modo equivalente, della resistenza) di un corpo immerso in flusso di Stokes. Queste simulazioni mostrano che, agendo su gradi di libertà opportunamente scelti, la parametrizzazione mediante *Free Form Deformation* è in grado di fornire una buona approssimazione della forma ottima ricavata analiticamente da Pironneau, con la stessa rapidità di convergenza mostrata dalla tecnica di *local boundary variation*.

Sono stati infine presentati i risultati dell'applicazione principale di questa Tesi, cioè la ricerca del *design* ottimale di un bypass aorto-coronarico. Come descritto nel CAPITOLO 1, si tratta di un problema piuttosto complesso, che ben si presta all'applicazione degli strumenti dell'ottimizzazione di forma: numerosi studi hanno infatti mostrato come le caratteristiche geometriche dell'innesto abbiano forte influenza sulle quantità fluidodinamiche, e come alcune di queste quantità (energia dissipata, vorticità, sforzi di taglio a parete, ...) siano correlate al processo di ristenosi, a causa del quale l'effetto dell'intervento può essere compromesso. Sono stati illustrati i risultati ottenuti minimizzando l'energia dissipata in un modello semplificato, in cui il flusso nelle coronarie è governato da un problema di Stokes stazionario e non stazionario (per tenere conto della natura pulsatile del flusso sanguigno). Le forme ottime così ottenute sono state poi validate mediante una simulazione con un modello di Navier-Stokes non stazionario, e confrontate con lavori precedenti: questa analisi ha permesso di concludere che sono preferibili geometrie in cui l'innesto è suturato ad una maggiore distanza dall'occlusione e forma angoli piccoli con l'arteria. Numerosi approfondimenti di questa analisi sono possibili; ad esempio:

- è possibile considerare geometrie più realistiche: le dimensioni nei pressi dell'anastomosi prossimale (suturata sull'aorta ascendente) e nei pressi dell'anastomosi distale (suturata alla coronaria occlusa) sono in realtà notevolmente differenti; l'anatomia delle coronarie è non planare e più complessa di una configurazione cilindrica, così come la stenosi risulta raramente simmetrica. Idealmente, la geometria da cui partire per risolvere il problema di ottimizzazione di forma dovrebbe essere ricostruita con opportune procedure di *medical imaging* eseguite sul singolo paziente, dal quale misurare anche il flusso di velocità da assegnare al profilo di inlet. A maggior ragione, con una geometria così complessa, si manifesta la validità di una parametrizzazione efficiente e di bassa dimensionalità, come la *Free Form Deformation*;
- è possibile risolvere il problema di ottimizzazione di forma direttamente sul modello emodinamico completo (problema di Navier-Stokes non stazionario): nella discussione dei risultati è stato mostrato infatti che la natura non lineare si manifesta in particolare in tre regioni (tratto di coronaria compreso tra l'innesto prossimale e la stenosi, tratto compreso tra la stenosi e l'innesto distale, tratto a valle dell'innesto distale), in corrispondenza delle quali sarebbe opportuno un ulteriore raffinamento della griglia di calcolo. Ulteriori sviluppi possono riguardare la scelta di una descrizione emodinamica ancora più realistica, considerando ad esempio effetti di interazione fluido-struttura o modelli reologici non-Newtoniani;
- è possibile ottimizzare rispetto ad un diverso funzionale costo, in particolare cercando di minimizzare la misura delle superfici a basso WSS o ad alto OSI; ciò permette da un lato di stabilire l'effetto che il funzionale costo ha sulla forma ottima e, dall'altro, di studiare quale indice sia più adeguato per le applicazioni mediche.



## Appendice A

# Implementazione in C++

In questa appendice viene presentata l'implementazione in C++ e *LifeV* di una libreria per problemi di ottimizzazione di forma in cui la parametrizzazione mediante la mappa *Free Form Deformation* (Sezioni 2.3.1 e 4.4.1), ed il problema di ottimizzazione rispetto ai gradi di libertà della mappa è risolto con il metodo del gradiente con proiezione (Sezioni 4.4.2 e 4.4.3). L'implementazione della mappa *Free Form Deformation* è descritta nell'Appendice A.1; l'implementazione del metodo del gradiente con proiezione è presentata nell'Appendice A.2. La struttura dati per la risoluzione di un generico problema di ottimizzazione di forma è illustrata nell'Appendice A.3; infine, si mostrerà nell'Appendice A.4 come utilizzare questa libreria per la risoluzione del problema presentato nella Sezione 5.1.1.

### A.1 La mappa *Free Form Deformation*

La classe `FreeFormDeformation` è ottenuta utilizzando il principio del *contenimento* di ulteriori classi ausiliarie, che vengono illustrate di seguito.

#### A.1.1 Classe `Displacement`

La dichiarazione della classe `Displacement` è la seguente:

```
1 class Displacement {
2 public:
3     Displacement();
4     virtual ~Displacement();
5     bool isEnabled() const;
6     LifeV::Real value() const;
7     void enable();
8     void disable();
9     void value(LifeV::Real value);
10    friend std::ostream &operator<<(std::ostream &output, const Displacement &D);
11 private:
12    bool M_enabled;
13    LifeV::Real M_value;
14 };
```

La classe `Displacement` rappresenta lo spostamento, in una fissata direzione coordinata, a cui un punto di controllo della *Free Form Deformation* può essere sottoposto. L'attributo privato `M_enabled` indica se tale spostamento è consentito e, in caso affermativo, il numero reale `M_value` indica l'entità della deformazione.

Il costruttore inizializza `M_enabled` a `false`. Vengono messi a disposizione gli *accessor* e i *mutator* per agire sullo stato interno della classe.

### A.1.2 Classe `ControlPoint`

La dichiarazione della classe `ControlPoint` è la seguente:

```

1 typedef std::vector<LifeV::UInt> ControlPointID;
2
3 class ControlPoint{
4 public:
5     ControlPoint(ControlPointID ID);
6     virtual ~ControlPoint();
7     const ControlPointID ID() const;
8     const LifeV::UInt numberOfEnabledDisplacements() const;
9     const LifeV::UInt spaceDimension() const;
10    Displacement& operator[](const LifeV::UInt& component);
11    const Displacement& operator[](const LifeV::UInt& component) const;
12    friend std::ostream &operator<<(std::ostream &output, const ControlPoint &CP);
13 private:
14    ControlPointID M_ID;
15    std::vector<Displacement> M_displacements;
16 };

```

La classe `ControlPoint` rappresenta un singolo punto di controllo. La  $i$ -esima componente di `M_ID` è utilizzata per specificare la propria posizione nella  $i$ -esima direzione della griglia di punti di controllo (ciò è sufficiente perché nella *Free Form Deformation* si ipotizza che i punti di controllo siano uniformemente distribuiti nel cubo  $[0, 1]^n$ , altrimenti oltre all'identificatore sarebbe necessario specificare anche l'effettiva posizione). La  $i$ -esima componente del vettore `M_displacements` rappresenta lo spostamento nella direzione  $i$ -esima.

Il costruttore inizializza il vettore degli identificatori per copia, disabilitando gli spostamenti in ciascuna direzione. Non vengono forniti metodi per modificare il vettore di identificatori, che quindi rimarrà costante durante l'esecuzione del programma, mentre, tramite l'**operator** `[]`, combinato con i metodi pubblici di `Displacement`, è possibile controllare lo stato degli spostamenti in ciascuna direzione.

### A.1.3 Classe `FFDReferenceMesh<RegionMesh>`

```

1 typedef std::vector<Domain1D> BoundingBox;
2
3 template <class RegionMesh>
4 class FFDReferenceMesh {
5 public:
6     FFDReferenceMesh();
7     FFDReferenceMesh(boost::shared_ptr<LifeV::MeshPartitioner<RegionMesh> >
8         OmegaPartitioner, BoundingBox D);
9     void setup(boost::shared_ptr<LifeV::MeshPartitioner<RegionMesh> > OmegaPartitioner
10         , BoundingBox D);
11     virtual ~FFDReferenceMesh();
12     const BoundingBox boundingBox() const;
13     const std::vector<typename RegionMesh::point_Type> & omegaPoints() const;
14     const LifeV::UInt spaceDimension() const;
15 private:
16     std::vector<typename RegionMesh::point_Type> M_Omega_Points;
17     BoundingBox M_D;
18 };

```

La classe *template* `FFDReferenceMesh<RegionMesh>` si occupa di incapsulare la struttura dati del dominio di riferimento associato alla *Free Form Deformation*. Essa viene inizializzata tramite il costruttore oppure tramite il metodo `setup`, a cui viene passato

1. un puntatore al `MeshPartitioner` della triangolazione del dominio di riferimento  $\Omega$ , il cui tipo è lasciato come *template* e può essere scelto tra quelli disponibili in *LifeV* (in particolare `RegionMesh3D` o, per applicazioni bidimensionali, `RegionMesh2D`),

2. il parallelepipedo  $D$  (in seguito denominato “*bounding box*”), di cui la triangolazione è un sottoinsieme.

Si sottolinea che:

- per quanto riguarda la triangolazione, gli unici elementi effettivamente memorizzati sono i vertici (variabile privata `M_Omega_Points`), perché questa informazione è sufficiente per deformare la mesh secondo la *Free Form Deformation*, come verrà mostrato in seguito;
- non è stato effettuato alcun controllo per garantire che il dominio di riferimento sia effettivamente un sottoinsieme del parallelepipedo  $D$  (memorizzato nella variabile privata denominata `M_D`): la violazione di questo vincolo provoca il malfunzionamento del programma, che termina lanciando un’eccezione (non catturata) non appena venga richiesta la deformazione secondo la mappa *Free Form Deformation*, in quanto l’immagine della mappa  $\psi$  include strettamente il dominio computazionale  $\hat{D}$ , provocando un errore nella valutazione dei polinomi di Bernstein. Questa scelta è stata effettuata in previsione di concedere, in una futura versione della libreria, la possibilità di operare deformazioni solo un sottoinsieme del dominio di riferimento, garantendo cioè la facoltà di concentrare i punti di controllo solo nella regione soggetta a controllo.

#### A.1.4 Classe **FFDMeshData**

La dichiarazione della classe `FFDMeshData` è la seguente:

```

1 class FFDMeshData: public LifeV::MeshData {
2 public:
3     FFDMeshData();
4     FFDMeshData( const GetPot& dataFile, const std::string& section_meshData = "
        space_discretization", const std::string& section_simulationType = "
        simulation_type");
5     FFDMeshData( const FFDMeshData& meshData );
6     virtual ~FFDMeshData();
7     void setup( const GetPot& dataFile, const std::string& section_meshData = "
        space_discretization", const std::string& section_simulationType = "
        simulation_type");
8     virtual void showMe( std::ostream& output = std::cout ) const;
9     std::string boundingBox() const;
10    bool isVTKImporterEnabled() const;
11    void enableVTKImporter();
12    void disableVTKImporter();
13    std::string FFDGridSize() const;
14    std::string VTKDir() const;
15    std::string VTKFile() const;
16    std::string simulationType() const;
17    bool & isLeader();
18    bool isLeader() const;
19 private:
20    std::string M_boundingBox;
21    std::string M_FFDGridSize;
22    bool M_enableVTKImporter;
23    std::string M_vtk_dir;
24    std::string M_vtk_file;
25    std::string M_simulationType;
26    bool M_isLeader;
27 };

```

La classe `FFDMeshData` è un’estensione, con ereditarietà pubblica, di `MeshData`, incluso nella libreria *LifeV*. Rispetto alla classe base, che si occupa sostanzialmente di ricavare informazioni da un oggetto `GetPot` sugli attributi della mesh (come ad esempio il formato e il nome del

file da importare), essa fornisce le ulteriori informazioni necessarie per la creazione di un oggetto `FreeFormDeformation`. Considerando ad esempio una simulazione tridimensionale, la struttura della sezione letta dal file `GetPot` è la seguente:

```

1 [./space_discretization]
2   mesh_dir      = ./data/Mesh/
3   mesh_file     = mesh.vol
4   mesh_type     = .vol
5   verbose      = true
6   boundingBox   = x_min,x_max,y_min,y_max,z_min,z_max
7   FFDGridSize  = K,L,M
8   enableVTKImporter = true
9   vtk_dir      = ./data/restarter/
10  vtk_file     = solution_restarter.vtk
11 [../]

```

intendendo  $x_{\min}, x_{\max}, y_{\min}, y_{\max}, z_{\min}, z_{\max}$  e  $K, L, M$  in accordo con i simboli utilizzati nella Sezione 2.3.1.

### A.1.5 Classe `Bernstein1D`

La dichiarazione della classe `Bernstein1D` è la seguente:

```

1 struct Bernstein1D {
2   static LifeV::UInt binomial(LifeV::UInt N, LifeV::UInt n);
3   static LifeV::Real bernstein(LifeV::UInt N, LifeV::UInt n, LifeV::Real x);
4   static LifeV::Real bernsteinDerivative(LifeV::UInt N, LifeV::UInt n, LifeV::Real x
5   );
6 };

```

La classe `Bernstein1D` fornisce metodi statici per la valutazione del coefficiente binomiale, dei polinomi di Bernstein monodimensionali e del gradiente di tali polinomi nel dominio monodimensionale  $[0, 1]$ .

### A.1.6 Classe `FreeFormDeformation<RegionMesh>`

La classe *template* `FreeFormDeformation<RegionMesh>` utilizza tutte le strutture dati precedenti per la creazione della mappa di deformazione dal dominio di riferimento  $\Omega$  a quello perturbato  $\Omega_o(\boldsymbol{\mu})$ . Il parametro *template* è nuovamente dato dal tipo della mesh.

La dichiarazione della classe `FreeFormDeformation<RegionMesh>` viene suddivisa raggruppando i metodi e gli attributi secondo il rispettivo compito.

#### Attributi privati e costruttori

```

1 public:
2   typedef FFDMeshData Data;
3   typedef RegionMesh RegionMeshType;
4   FreeFormDeformation(boost::shared_ptr<RegionMesh> referenceDomain, const
5     BoundingBox & boundingBox, const FFDGridSize & grid_size, const boost::
6     shared_ptr<Epetra_Comm> & commPtr);
7   FreeFormDeformation(const FFDMeshData & meshData, const boost::shared_ptr<
8     Epetra_Comm> & commPtr);
9   virtual ~FreeFormDeformation();
10 private:
11  boost::shared_ptr<Epetra_Comm> M_commPtr;
12  boost::shared_ptr<LifeV::MeshPartitioner<RegionMesh> > M_deformedDomainPartitioner
13  ;
14  FFDReferenceMesh<RegionMesh> M_referenceDomain;
15  FFDGridSize M_grid_size;
16  std::vector<ControlPoint> M_grid;

```

```
13 void GridConstructor();
```

L'implementazione di un oggetto `FreeFormDeformation<RegionMesh>` è basata su:

1. un dominio di riferimento `M_referenceDomain` di tipo `FFDReferenceMesh<RegionMesh>`;
2. uno *shared pointer* `M_deformedDomainPartitioner` al partizionatore della mesh del dominio deformato;
3. il comunicatore `Epetra_Comm` che è stato utilizzato per partizionare la mesh nella configurazione di riferimento;
4. un vettore di numeri naturali `M_grid_size`, la cui componente  $i$ -esima memorizza il numero di punti di controllo nella corrispondente direzione, diminuito di 1;
5. un vettore `M_grid` di `ControlPoint`, in cui sono memorizzati tutti i punti di controllo della mappa.

Il metodo ausiliario `void GridConstructor()`, che inizializza la griglia dei punti di controllo ad una configurazione “di default” (senza alcuno spostamento abilitato), è utilizzato in entrambi i costruttori:

1. `FreeFormDeformation(boost::shared_ptr<RegionMesh> referenceDomain, const BoundingBox & boundingBox, const FFDGridSize & grid_size, const boost::shared_ptr<Epetra_Comm> & commPtr)` inizializza i membri privati con i corrispondenti dati e genera la griglia “di default”.
2. `FreeFormDeformation(const FFDMeshData & meshData, const boost::shared_ptr<Epetra_Comm> & commPtr)` si occupa invece di leggere la mesh di riferimento (e quella deformata, che viene inizializzata pari ad essa) grazie alle opportune funzioni messe a disposizione da *LifeV*, e di consultare `meshData` per decidere come generare la griglia dei punti di controllo. In particolare:
  - (a) se l'attributo `enableVTKImporter` del file di dati *GetPot* è impostato a `true`, allora i punti di controllo sono importati dal file presente `vtk_dir/vtk_file`, in cui è memorizzata la posizione, le direzioni abilitate e l'entità dello spostamento in ciascuna direzione;
  - (b) se non è stato possibile effettuare l'importazione, allora vengono utilizzati gli attributi `boundingBox` e `FFDGridSize` del file *GetPot*, e crea la griglia “di default”.

## Gestione della deformazione

```
1 public:
2   const LifeV::UInt spaceDimension() const;
3   const LifeV::UInt gridSize(LifeV::UInt i) const;
4   const LifeV::UInt boundingBoxEdgeLength(LifeV::UInt i) const;
5   const LifeV::UInt numberOfEnabledControlPoints() const;
6   VectorEnabledControlPoints_ptr enabledControlPoints();
7   const LifeV::UInt numberOfEnabledDisplacements() const;
8   VectorEnabledDisplacements_ptr enabledDisplacements();
9   ControlPoint& operator[](const ControlPointID & controlPointID);
10  LifeV::Real bernsteinOnReferenceDomain(const SimplePoint & x, const ControlPointID
    & controlPointID) const;
11  Eigen::Matrix<LifeV::Real, Eigen::Dynamic, 1> bernsteinGradientOnReferenceDomain(
    const SimplePoint & x, const ControlPointID & controlPointID) const;
12  SimplePoint operator()(const SimplePoint & x) const;
```

```

13 Eigen::Matrix<LifeV::Real,Eigen::Dynamic,Eigen::Dynamic> jacobian(const
    SimplePoint & x) const;
14 Eigen::Matrix<LifeV::Real,Eigen::Dynamic,Eigen::Dynamic> invjacobian(const
    SimplePoint & x) const;
15 void deformReferenceDomain();
16 boost::shared_ptr<LifeV::MeshPartitioner<RegionMesh> > deformedDomainPartitioner()
    const;
17 void checkDeformedDomain() const;
18 private:
19 LifeV::UInt inGridID(const ControlPointID& controlPointID) const;
20 SimplePoint psi(SimplePoint x) const;
21 SimplePoint invpsi(SimplePoint x) const;
22 SimplePoint operator()(const ControlPointID & controlPointID, bool
    disableDisplacements) const;
23 SimplePoint operator()(const typename RegionMesh::point_Type & p) const;

```

avendo definito i seguenti tipi

```

1 typedef std::vector<LifeV::UInt> FFDGridSize;
2 typedef std::vector<LifeV::Real> SimplePoint;
3 typedef std::vector<ControlPoint_ptr> VectorEnabledControlPoints_ptr;
4 typedef std::vector<Displacement_ptr> VectorEnabledDisplacements_ptr;

```

I principali metodi per la gestione della deformazione sono:

- o `VectorEnabledControlPoints_ptr enabledControlPoints()` , che permette di estrarre un vettore di puntatori ai punti di controllo con almeno una componente di spostamento abilitata;
- o `VectorEnabledDisplacements_ptr enabledDisplacements()` , che permette di estrarre un vettore di puntatori alle singole componenti di spostamento abilitate, senza però tenere traccia del punto di controllo a cui sono associate. Con la notazione introdotta nella Sezione 2.3.1, esso restituisce il vettore  $\mu$ ;
- o `ControlPoint& operator[] (const ControlPointID & controlPointID)` , il quale si occupa di restituire una referenza al punto di controllo con identificatore pari a `controlPointID` , in modo che sia possibile modificare gli attributi della descrizione parametrica del dominio;
- o `SimplePoint operator() (const SimplePoint & x) const` , il quale restituisce, dato un punto `x` nella configurazione di riferimento, la sua posizione nella configurazione deformata;
- o `void deformReferenceDomain()` , che deforma il dominio di riferimento secondo la *Free Form Deformation*, aggiornando la posizione dei vertici della mesh.

```

1 template <class RegionMesh>
2 void FreeFormDeformation<RegionMesh>::deformReferenceDomain() {
3     std::vector<typename RegionMesh::point_Type> refPointList(M_referenceDomain.
        omegaPoints());
4     for (LifeV::UInt i = 0; i < refPointList.size(); ++i) {
5         SimplePoint displaced_p_vec(this->operator () (refPointList[i]));
6         // copy back to the deformed mesh (via point_Type)
7         for (LifeV::UInt c = 0; c < spaceDimension(); ++c)
8             M_deformedDomainPartitioner->meshPartition()->point(i).coordinate(c) =
                displaced_p_vec[c];
9     }
10 }

```

Lo *shared pointer* al partizionatore della mesh deformata è restituito con `boost::shared_ptr<LifeV::MeshPartitioner<RegionMesh> > deformedDomain() const` ;

- **void** `checkDeformedDomain()` **const** , il quale lancia un'eccezione se la deformazione ha corrotto la mesh, cioè ha generato elementi dal volume negativo.

### Esportazione per *ParaView* e file di re-inizializzazione

```

1 public:
2     template <class RegionMeshb> friend std::ostream& operator<<(std::ostream &output,
3         const FreeFormDeformation<RegionMeshb> &FFD);
4     void exportVisualizationVTK(std::string filename);
5     void exportRestarterVTK(std::string filename);
6     void exportRestarterVTK(const FFDMeshData & meshData);
7     void exportEnabledDisplacementsCustomDataVTK(std::string filename, const
8         VectorEnabledControlPoints_ptr & enabledControlPoints, const Eigen::Matrix<
9         LifeV::Real,Eigen::Dynamic,1> & data);
10 private:
11     void exportVTK(std::string filename, bool disableDisplacements);

```

I principali metodi che gestiscono l'esportazione sono:

- **void** `exportVisualizationVTK(string filename)` , il quale esporta in formato compatibile con il visualizzatore *ParaView* lo stato completo dei punti di controllo nella posizione deformata;
- **void** `exportRestarterVTK(string filename)` , che esporta in formato compatibile con il visualizzatore *ParaView* lo stato completo dei punti di controllo, mantenendoli però nella loro posizione di riferimento in modo che sia possibile importare la griglia in una successiva simulazione.

## A.2 Metodo del gradiente con proiezione

In questa sezione viene presentata l'implementazione nella classe C++ `ProjectionMethod` del metodo del gradiente con proiezione per un problema di ottimizzazione su un insieme  $M \subset \mathbb{R}^p$  individuato da vincoli di *bound*, vincoli lineari e vincoli non lineari.

### A.2.1 Classe `Function`

La dichiarazione della classe `Function` è la seguente:

```

1 struct Function {
2     Function();
3     virtual Real operator() (const Vector & x) = 0;
4     virtual Vector gradient(const Vector & x) = 0;
5 };

```

Essa fornisce la struttura di una generica funzione scalare e del suo gradiente.

### A.2.2 Classi per la gestione di problemi vincolati

Il seguente **enum** fornisce il tipo di vincolo: uguaglianza ( $g(x) = 0$ ) oppure disuguaglianza ( $g(x) \geq 0$ )

```

1 enum ConstraintType {
2     equality = 0,
3     inequality = 1
4 };

```

Sono disponibili tre classi di vincoli:

1. vincoli di *bound*, che fanno uso della seguente struttura dati:

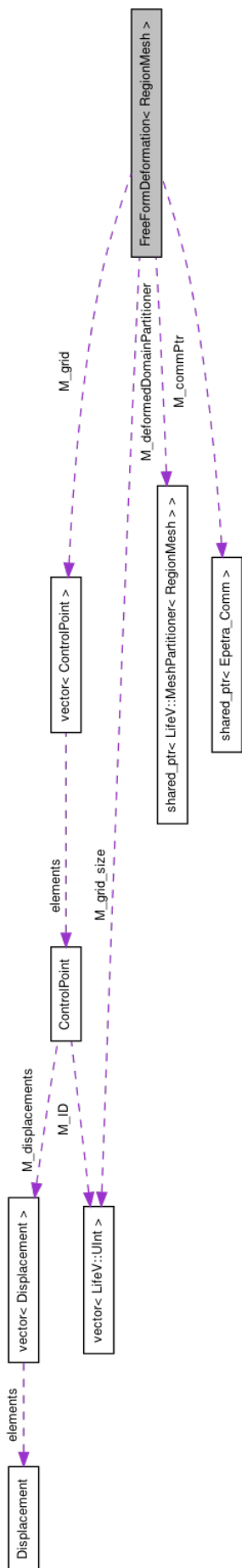


Figura A.1: Collaboration diagram della classe FreeFormDeformation<RegionMesh> .



```

1 struct Bounds {
2     Vector lowerBounds;
3     Vector upperBounds;
4 };

```

2. vincoli lineari  $A\mathbf{x} = \mathbf{b}$  oppure  $A\mathbf{x} \geq \mathbf{b}$ . La struttura di ciascun vincolo  $\mathbf{a}_i \cdot \mathbf{x} = b_i$  oppure  $\mathbf{a}_i \cdot \mathbf{x} \geq b_i$  è

```

1 struct LinearConstraint {
2     ConstraintType constraintType;
3     Vector coefficients;
4     Real RHS;
5 };

```

La valutazione della funzione  $\mathbf{a}_i \cdot \mathbf{x} - b_i$  e del suo gradiente  $\mathbf{a}_i$  è effettuata tramite la classe `LinearConstraint_Function`, derivata da `Function`.

```

1 class LinearConstraint_Function: public Function {
2 public:
3     LinearConstraint_Function(Vector coefficients, Real RHS);
4     Real operator() (const Vector & x);
5     Vector gradient(const Vector & x);
6 private:
7     Vector M_coefficients;
8     Real M_RHS;
9 };

```

3. vincoli non lineari  $\mathbf{g}(\mathbf{x}) = \mathbf{0}$  oppure  $\mathbf{g}(\mathbf{x}) \geq \mathbf{0}$ . Ciascun vincolo scalare  $g_i(\mathbf{x}) = 0$  o  $g_i(\mathbf{x}) \geq 0$  è rappresentato da

```

1 struct NonlinearConstraint {
2     ConstraintType constraintType;
3     boost::shared_ptr<Function> constraintFunction;
4 };

```

### A.2.3 Classe `Exporter`

La dichiarazione della classe `Exporter` è la seguente:

```

1 struct Exporter {
2     Exporter();
3     virtual void operator() (const Result & min, const std::vector<Real> &
4         backtrackingStepSizes, const std::vector<Real> &
5         backtrackingIntermediateFunctionValues, const std::vector<Real> &
6         stoppingCriteriaValues, const Vector & gradient, const UInt & iterationNumber
7     );
8 };

```

Essa può essere utilizzata per esportare i risultati ottenuti all'iterazione  $n$ -esima, insieme con tutte le iterazioni di *backtracking* che sono state necessarie per il calcolo del passo  $\alpha^{(n)}$ .

### A.2.4 Classe `StoppingCriteria`

La dichiarazione della classe `StoppingCriteria` è la seguente:

```

1 class StoppingCriteria {
2 public:
3     enum criteria_t {
4         RelativeToleranceOnCost = 0,
5         AbsoluteToleranceOnCost,
6         CostValue,

```

```

7     AbsoluteToleranceOnGradient,
8     RelativeToleranceOnVariablesNorm,
9     AbsoluteToleranceOnVariablesNorm,
10    MaximumIterations
11 };
12 StoppingCriteria(criteria_t criteria=RelativeToleranceOnCost, Real value=0.0);
13 StoppingCriteria(std::string criteria, Real value);
14 Real value() const;
15 std::string toString() const;
16 Real operator()(Result previous, Result current, Vector gradient_current, UInt
    iterationNumber) const;
17 bool converged(Result previous, Result current, Vector gradient_current, UInt
    iterationNumber) const;
18 private:
19     criteria_t M_criteria;
20     Real M_value;
21 };

```

La classe `StoppingCriteria` fornisce la gestione dei dati necessari alla definizione di un criterio d'arresto: in particolare sono contemplati criteri d'arresto basati sulla tolleranza assoluta o relativa (all'iterazione precedente) rispetto al valore del funzionale costo, alla norma del gradiente o alla norma delle variabili soggette ad ottimizzazione. È inoltre presente un criterio per limitare il numero di massimo di iterazioni.

### A.2.5 Classe `BacktrackingData`

La dichiarazione della classe `BacktrackingData` è la seguente:

```

1 struct BacktrackingData {
2     Real ArmijoConstant;
3     Real backtrackingScale;
4     Real initialStepsize;
5     bool useStaticStepsize;
6     UInt maxBacktrackingIterations;
7     BacktrackingData(Real ArmijoConstant = 0.1, Real backtrackingScale = 0.5, Real
        initialStepsize = 1.0, bool useStaticStepsize = false, UInt
        maxBacktrackingIterations = std::numeric_limits<UInt>::max());
8 };

```

Gli attributi reali memorizzati nelle variabili `ArmijoConstant`, `backtrackingScale` e `initialStepsize` rappresentano rispettivamente le costanti  $c$ ,  $\rho$  e  $\bar{\alpha}$  della (4.9)-(4.10).

Si è ritenuto necessario, però, modificare il comportamento dell'algoritmo standard di *backtracking*, anzitutto garantendo la possibilità di imporre un numero massimo di iterazioni, memorizzato in `maxBacktrackingIterations`.

Inoltre, considerato l'onere computazionale di ogni valutazione del funzionale costo nel caso di problemi di ottimizzazione di forma, se `useStaticStepsize` è impostato a `true` il procedimento di *backtracking* viene modificato in modo che il valore di  $\bar{\alpha}$  sia  $\bar{\alpha} = \bar{\alpha}(n)$ , essendo  $n$  l'indice sulle iterazioni del metodo del gradiente. In particolare si pone  $\bar{\alpha}(n)$  uguale al passo ottimo  $\alpha^{(n-1)}$  individuato all'iterazione precedente del metodo del gradiente con proiezione.

### A.2.6 Classe `ProjectionMethodData`

La dichiarazione della classe `ProjectionMethodData` è la seguente:

```

1 struct ProjectionMethodData {
2     std::vector<StoppingCriteria> stoppingCriteria;
3     BacktrackingData backtrackingData;
4     Real tolNewtonRaphson;
5     bool verbose;

```

```

6   ProjectionMethodData(std::vector<StoppingCriteria> stoppingCriteria=std::vector<
      StoppingCriteria>(1, StoppingCriteria()), BacktrackingData backtrackingData =
      BacktrackingData(), Real tolNewtonRaphson = 0.01, bool verbose = true);
7   ProjectionMethodData(std::string GetPotFileName, std::string GetPotSection="
      optimizer");
8   virtual ~ProjectionMethodData();
9 };

```

I due oggetti principali memorizzati nella classe `ProjectionMethodData` sono un vettore di `StoppingCriteria`, in modo che il metodo del gradiente sia terminato quando almeno uno di essi è verificato, ed un elemento di tipo `backtrackingData`.

È possibile, oltre che per copia, inizializzare tali dati tramite la lettura di un oggetto *GetPot*, formattato come nell'esempio seguente:

```

1  [./optimizer]
2    [./stopping_criteria]
3      criteria_0 = AbsoluteToleranceOnGradient
4      value_0 = 0.001
5      criteria_1 = MaximumIterations
6      value_1 = 15
7    [../]
8
9    [./backtracking]
10     ArmijoConstant = 0.5
11     backtrackingScale = 0.5
12     initialStepsize = 11
13     useStaticStepsize = true
14     maxBacktrackingIterations = 10
15   [../]
16
17   tolNewtonRaphson = 0.01 # for (linear and) nonlinear constraints
18   verbose = true
19 [../]

```

### A.2.7 Classe `ProjectionMethod`

La dichiarazione della classe `ProjectionMethod`, che implementa il metodo del gradiente con proiezione per un problema di minimizzazione vincolata, viene suddivisa nelle seguenti sezioni, in modo da riordinare metodi ed attributi secondo il rispettivo compito.

#### Costruttore e dati del metodo

```

1 public:
2   ProjectionMethod(const ProjectionMethodData & projectionMethodData);
3   virtual ~ProjectionMethod();
4 private:
5   ProjectionMethodData M_projectionMethodData;

```

Il costruttore inizializza l'attributo `M_projectionMethodData`, contenente i dati su criterio di arresto e procedura di *backtracking*: le altre informazioni (funzione obiettivo, vincoli, ...) devono essere fornite tramite gli opportuni metodi pubblici che verranno presentati nel seguito.

#### Gestione della funzione costo

```

1 public:
2   void setCostFunction(const boost::shared_ptr<Function> & costFunction);
3 private:
4   boost::shared_ptr<Function> M_costFunction;

```

Prima di lanciare l'algoritmo di ottimizzazione è necessario impostare quale sia la funzione costo da minimizzare tramite il metodo `setCostFunction`.

### Gestione dei vincoli

```

1 public:
2     void setBounds(const Vector & lowerBounds, const Vector & upperBounds);
3     void setBounds(const Bounds & bounds);
4     void addLinearConstraint(const ConstraintType & constraintType, const Vector &
      coefficients, const Real & RHS);
5     void addLinearConstraint(const LinearConstraint & linearConstraint);
6     void addNonlinearConstraint(const ConstraintType & constraintType, const boost::
      shared_ptr<Function> & constraintFunction);
7     void addNonlinearConstraint(const NonlinearConstraint & nonlinearConstraint);
8     void setTolNewtonRaphson(const Real & tolNewtonRaphson=0.01);
9 private:
10    Vector M_lowerBounds;
11    Vector M_upperBounds;
12    std::vector<ConstraintType> M_linearANDnonlinearConstraint_Types;
13    std::vector<boost::shared_ptr<Function> > M_linearANDnonlinearConstraint_Functions
      ;

```

L'interfaccia pubblica mette a disposizione vari metodi per aggiungere i vincoli scalari delle tre classi mostrate in precedenza (*bound*, lineari, non lineari). Va sottolineato che vincoli lineari e non lineari sono memorizzati nello stesso vettore: questa scelta è stata effettuata perché, per quanto riguarda l'operatore di proiezione, per entrambi si può utilizzare la formula (4.15) (con la differenza che per i vincoli lineari il metodo di Newton-Raphson converge in una iterazione alla proiezione esatta, mentre per vincoli non lineari ciò, in generale, non vale).

### Gestione della proiezione sui vincoli

```

1 private:
2     class M_projection_Base {
3     public:
4         M_projection_Base();
5         M_projection_Base(ProjectionMethod* fake_this);
6         virtual Vector operator()(const Vector & x) = 0;
7     protected:
8         ProjectionMethod* fake_this;
9     };
10    friend class M_projection_Base;
11    struct M_projection_Bounds: public M_projection_Base {
12        M_projection_Bounds(ProjectionMethod* fake_this);
13        virtual Vector operator()(const Vector & x);
14    };
15    friend struct M_projection_Bounds;
16    struct M_projection_LinearANDNonlinearConstraints: public M_projection_Base {
17        M_projection_LinearANDNonlinearConstraints(ProjectionMethod* fake_this);
18        virtual Vector operator()(const Vector & x);
19    };
20    friend struct M_projection_LinearANDNonlinearConstraints;
21    std::vector<boost::shared_ptr<M_projection_Base> > M_iterativeProjections;
22    Vector projection(const Vector & x);

```

La classe astratta `M_projection_Base` fornisce la struttura di un generico operatore di proiezione: il metodo `Vector operator()(const Vector & x)` restituisce  $\mathbf{x}_P = \mathbf{P}(\mathbf{x})$ , ed i dati dei vincoli sui quali effettuare la proiezione sono accessibili direttamente dall'istanza di `ProjectionMethod`, accessibile attraverso il puntatore `fake_this`.

L'operatore (4.11) per un problema con soli vincoli di *bound* è implementato nella classe `M_projection_Bounds`; alla prima chiamata di uno dei due metodi `setBounds()` un oggetto di tipo `M_projection_Bounds` viene aggiunto al vettore `M_iterativeProjections`.

Il metodo di Newton-Raphson (4.15) per problemi con soli vincoli non lineari di uguaglianza è implementato nella classe `M_projection_LinearANDNonlinearConstraints` (come già accennato in precedenza esso si riduce a (4.12) nel caso di soli vincoli lineari); alla prima chiamata di uno dei due metodi `addLinearConstraint()` o di uno dei due metodi `addNonlinearConstraint()`, in coda all'attributo privato `M_iterativeProjections` viene aggiunto un oggetto di tipo `M_projection_LinearANDNonlinearConstraints`.

Infine, il metodo `Vector projection(const Vector & x)` si occupa di effettuare la proiezione sull'insieme definito da tutti i vincoli, consultando il vettore `M_iterativeProjections`: nel caso esso sia vuoto (problema non vincolato), restituisce `x`; nel caso contenga un solo elemento (problema con vincoli di una sola classe: o solo vincoli di *bound* o solo vincoli lineari/non lineari), chiama l'**operator**() della corrispondente classe; nel caso contenga due elementi, applica a passi alterni i due operatori di proiezione fino a convergenza.

### Esportazione dei risultati

```
1 public:
2     void setExporter(const boost::shared_ptr<Exporter> & exporter);
3 private:
4     boost::shared_ptr<Exporter> M_exporter;
```

Qualora sia necessario esportare i risultati su file di testo è possibile creare una classe derivata da `Exporter` e sostituirla all'oggetto `M_exporter` di default (che non fa nulla) tramite il metodo `setExporter()`.

### I metodi principali

```
1 public:
2     Result minimize(const Vector & x0);
3     const UInt & functionEvaluationNumber();
4     const UInt & iterationNumber();
5 private:
6     UInt M_functionEvaluationNumber;
7     UInt M_iterationNumber;
8     bool converged(Result prevmin, Result min, Vector gradient);
9     struct M_BacktrackingOutput {
10         Result result;
11         std::vector<Real> stepsizes;
12         std::vector<Real> intermediateFunctionValues;
13     };
14     struct M_BacktrackingException: public std::runtime_error {
15         M_BacktrackingOutput invalidBacktrackingOutput;
16         explicit M_BacktrackingException(const std::string& what_arg,
17             M_BacktrackingOutput invalidBacktrackingOutput);
17     };
18     M_BacktrackingOutput backtracking(const Result & min, const Vector & gradient,
19         const Vector & searchDirection, const Real & norm_gradient_0th);
```

Il metodo `minimize()` è il metodo centrale della classe, cioè quello che esegue l'ottimizzazione a partire dal valore iniziale `x0`. Esso invoca ad ogni iterazione il metodo `backtracking()`, che esegue la ricerca monodimensionale inesatta con il procedimento di *backtracking* descritto dalle (4.9)-(4.10), opportunamente modificato per tenere conto delle proposte della Appendice A.2.5.

L'ottimizzazione è terminata quando il metodo `converged()` ritorna **true**, cioè quando uno tra i criteri d'arresto impostati in `M_projectionMethodData` è verificato, oppure a causa di un'eccezione di tipo `M_BacktrackingException`, lanciata quando al termine del ciclo di *backtracking* non sia stato trovato alcun valore  $\alpha_n$  che rispetti il criterio di Armijo.

Al termine della procedura di ottimizzazione è possibile ottenere sia il numero di iterazioni (i.e., il numero di valutazioni del gradiente della funzione costo) tramite `iterationNumber()`, sia il numero di valutazioni della funzione costo tramite il metodo `functionEvaluationNumber()` (solitamente `functionEvaluationNumber() >> iterationNumber()` a causa del ciclo di *backtracking*).

### A.3 Generico problema di ottimizzazione di forma

La struttura dati per la gestione di un problema di ottimizzazione di forma è basata sul *design pattern* del *template method*.

Infatti, trascurando momentaneamente le classi ausiliarie, la struttura dati è basata su tre classi astratte, in ordine crescente di “specializzazione”, a cui segue almeno una classe concreta:

1. la classe astratta base è `ShapeOptimizationProblem<ShapeParametrization>`, che vuole rappresentare un generico problema di *shape optimization*, con una parametrizzazione della forma gestita da un oggetto del tipo *template* `ShapeParametrization`. In questa classe sono dichiarati, come metodi *pure virtual*, i metodi comuni a tutti i problemi di ottimizzazione di forma (ad esempio un metodo per calcolare il valore del funzionale costo, un metodo per calcolare la soluzione del problema di stato, ...): essi dovranno essere implementati nella successive classi derivate. Le astrazioni presenti in questa classe<sup>1</sup> riguardano
  - (a) la parametrizzazione della forma;
  - (b<sub>1</sub>) l’espressione analitica del funzionale costo e del suo gradiente;
  - (b<sub>2</sub>) il calcolo del funzionale costo e del suo gradiente;
  - (c) la risoluzione approssimata del problema di stato e del problema aggiunto;
  - (d) la scelta del metodo di discesa utilizzato nella ricerca della forma ottima;
  - (e) la formulazione dei vincoli a cui la forma ottima è sottoposta.
2. il primo livello di ereditarietà è dato dalla classe astratta `FFDShapeOptimizationProblem<RegionMesh>`, derivata da `ShapeOptimizationProblem<FreeFormDeformation<RegionMesh>>`. Essa, dopo aver fissato il livello di astrazione (a) imponendo una parametrizzazione della forma secondo la mappa *Free Form Deformation*, è in grado di implementare i metodi che riguardano l’astrazione (b<sub>2</sub>), sfruttando in particolare la (4.6) per quanto riguarda il calcolo del gradiente del funzionale costo rispetto ai parametri della mappa *Free Form Deformation*.
3. il secondo livello di ereditarietà è dato dalla classe astratta `FFDProjectionMethodShapeOptimizationProblem<RegionMesh>`, derivata dalla classe `FFDShapeOptimizationProblem<RegionMesh>`. Essa implementa i metodi che riguardano l’astrazione (d), occupandosi di interfacciare la classe padre con i metodi della classe `ProjectionMethod` per la risoluzione, tramite il metodo del gradiente con proiezione, del problema di ottimizzazione rispetto alle variabili  $\mu$ .
4. l’ultimo livello di ereditarietà è quello che specifica il problema concreto, come verrà mostrato nella Appendice A.4. In esso vengono implementate le astrazioni (b<sub>1</sub>), (c) ed (e).

Tuttavia questo schema di ereditarietà non è pienamente soddisfacente per quanto riguarda l’astrazione (e): infatti, per lo meno nelle applicazioni illustrate in questa Tesi, il vincolo di volume

<sup>1</sup>Verrà presentata, nella Appendice A.3.4, l’implementazione di questa classe, e si potrà notare come alcuni di questi “livelli di astrazione” non siano associabili direttamente ad un gruppo di metodi *pure virtual*. Questo elenco è quindi da considerarsi come una “idealizzazione” che ha guidato la stesura del codice, anche se poi l’effettiva implementazione se ne discosta leggermente.

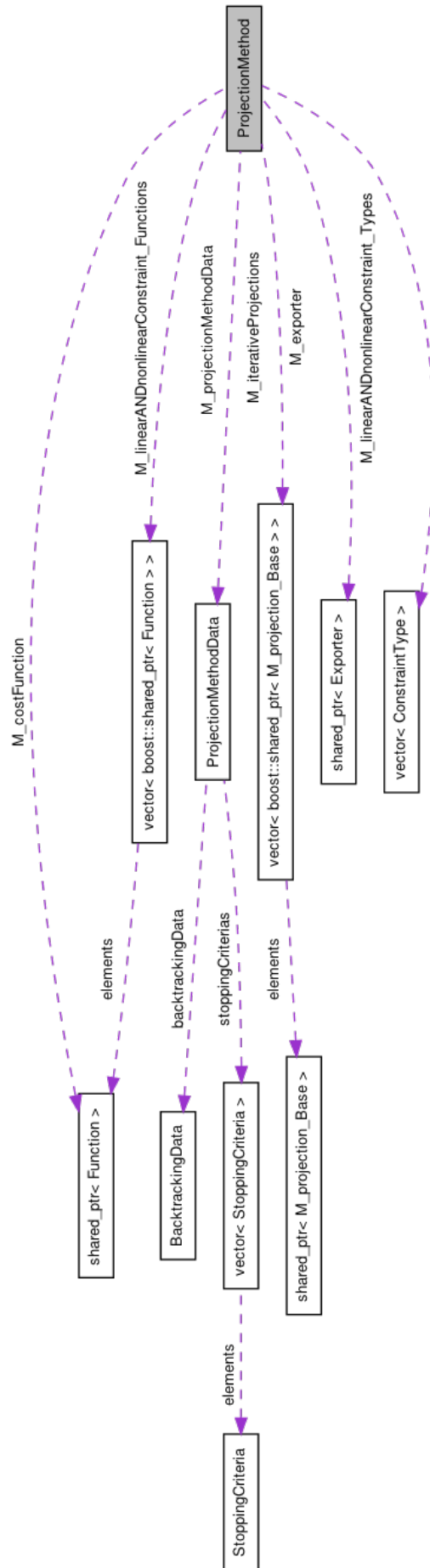


Figura A.2: Collaboration diagram della classe ProjectionMethod .

è presente in molti dei casi considerati. Si è deciso quindi – pur lasciando la possibilità a ciascuna classe concreta di specificare la propria formulazione dei vincoli – di implementare una volta per tutte i vincoli più comuni. Questo obiettivo è stato raggiunto tramite il *design pattern* del *decorator*: la classe `FFDProjectionMethodShapeOptimizationProblem_DecoratorBase` <RegionMesh> è la base di un decoratore grazie al quale sia possibile modificare (unicamente) la gestione dei vincoli. Da esso derivano `AddVolumeConstraintDecorator`<RegionMesh>, che aggiunge un vincolo di volume, e `FixPointConstraintDecorator`<RegionMesh>, che, assegnato un punto, garantisce che, per qualsiasi configurazione deformata, esso rimanga nella posizione occupata nella configurazione di riferimento.

### A.3.1 Classe `ShapeOptimizationData`

La dichiarazione della classe `ShapeOptimizationData` è la seguente:

```

1 struct ShapeOptimizationData {
2     GetPot dataFile;
3     std::string section_meshData;
4     std::string section_simulationType;
5     std::string section_exporter;
6     std::string section_timeDiscretization;
7     ShapeOptimizationData(std::string GetPotFileName="", std::string section_meshData=
8         "space_discretization", std::string section_simulationType="simulation_type",
9         std::string section_exporter="exporter", std::string
10        section_timeDiscretization="time_discretization");
11 virtual ~ShapeOptimizationData();
12 };

```

Questa classe si occupa di leggere da un file *GetPot* le informazioni necessarie per un generico problema di ottimizzazione di forma.

Per quanto riguarda la struttura delle sezioni dei file di dati, si ha che:

- o `section_meshData` è il nome della sezione che contiene informazioni sulla discretizzazione spaziale. Un esempio è stato riportato nella Appendice A.1.4 per quanto riguarda la parametrizzazione geometrica governata da *Free Form Deformation*;
- o `section_simulationType` è il nome della sezione che contiene un'informazione sul tipo di simulazione da lanciare. Ad esempio, nel caso di parametrizzazione geometrica data dalla *Free Form Deformation*, sono disponibili due tipi di modalità di esecuzione, come indicato nel seguente file *GetPot* :

```

1 [simulation_type]
2     type = optimize #optimize or startup
3 [../]

```

La modalità `startup`, da eseguire *una tantum*, si limita alla lettura dai dati della parametrizzazione geometrica e alla successiva esportazione su file della griglia di punti di controllo. Dopo aver modificato manualmente questo file (abilitando almeno uno spostamento, dato che originariamente sono tutti disabilitati!), è necessario lanciare nuovamente l'eseguibile in modalità `optimize` in modo che, se `enableVTKImporter` è `true`, la griglia e lo stato dei punti di controllo venga letta da file e venga avviato l'algoritmo di ottimizzazione;

- o `section_exporter` è il nome della sezione contiene i dati sull'esportazione dei risultati. Nel caso in cui il metodo di ottimizzazione numerica sia basato sulla classe `ProjectionMethod`, si può utilizzare il file seguente

```

1 [./exporter]
2     type      = hdf5
3     multimesh = false
4     start     = 0

```



```

5   save      = 1
6   post_dir  = ./
7   shapeParametrizationFileName = FFD
8   gradientFileName = gradient
9   costFunctionalGnuplotFileName = costFunctionalGnuplot
10  backtrackingHistoryFileName = backtrackingHistory
11  solutionFileName = solution
12  [../]

```

o `section_timeDiscretization` è il nome della sezione che contiene informazioni sulla discretizzazione temporale.

### A.3.2 Classi per la memorizzazione delle soluzioni dei problemi di stato ed aggiunto

```

1  typedef std::vector<boost::shared_ptr<LifeV::VectorEpetra> > Solutions;
2  typedef std::vector<Solutions> Solutions_VectorOverTime;
3  template <class RegionMesh>
4  struct FESpaces {
5      typedef std::vector<boost::shared_ptr<LifeV::FESpace<RegionMesh, LifeV::MapEpetra>
6      > > Type;
7  };

```

Le soluzioni dei problemi di stato ed aggiunto sono memorizzate in un oggetto di tipo `Solutions_VectorOverTime`. Esso è caratterizzato da due indici: il primo è l'indice dell'iterazione temporale, il secondo è indice sull'incognita (ad esempio, in un problema di Stokes si può arbitrariamente associare l'indice 0 alla velocità e l'indice 1 alla pressione).

La componente  $i$ -esima del vettore `FESpaces<RegionMesh>::Type` memorizza lo spazio ad elementi finiti a cui l'incognita  $i$ -esima appartiene: si suppone che lo stesso spazio ad elementi finiti sia condiviso tra una variabile di stato e la sua aggiunta.

### A.3.3 Classi per la gestione della dipendenza temporale

```

1  class TimeDependenceType {
2  public:
3      enum Type {
4          steady = 0,
5          unsteady,
6      };
7      TimeDependenceType & operator=(const TimeDependenceType::Type & type_enum);
8      bool operator==(const TimeDependenceType::Type & type_enum) const;
9      bool operator!=(const TimeDependenceType::Type & type_enum) const;
10 private:
11     Type type;
12 };

```

La classe `TimeDependenceType` definisce la dipendenza temporale del problema di ottimizzazione di forma: `steady` o `unsteady`. Per comodità essa fornisce inoltre gli operatori di assegnamento e confronto.

```

1  class TimeData: public LifeV::TimeData {
2  public:
3      TimeData();
4      TimeData( const GetPot& dataFile, const std::string& section = "
5          time_discretization" );
6      TimeData( const TimeData& TimeData);
7      LifeV::UInt endTime_StepNumber() const;
8      LifeV::Real timeAtIteration(LifeV::UInt iteration) const;
9  private:
10     void setupEndTime_StepNumber();

```

```

10     LifeV::UInt M_endTime_StepNumber;
11 };

```

La classe `TimeData` estende `LifeV::TimeData` fornendo due ulteriori metodi: il metodo `endTime_StepNumber()` per il calcolo dell'indice corrispondente all'ultima iterazione temporale, ed il metodo `timeAtIteration()` per calcolare l'istante di tempo associato all'iterazione `iteration`.

```

1  template <class ShapeOptimizationProblem_SubClass>
2  class TimeDependence {
3  public:
4      TimeDependence(ShapeOptimizationProblem_SubClass * fake_this);
5      TimeDependence & operator=(const TimeDependenceType::Type & type_enum);
6      const TimeDependenceType & type() const;
7      const TimeData & timeData() const;
8  private:
9      ShapeOptimizationProblem_SubClass * fake_this;
10     TimeDependenceType M_type;
11     TimeData M_timeData;
12 };

```

La classe `TimeDependence` gestisce la dipendenza temporale del problema di ottimizzazione di forma `fake_this` tramite l'`operator=`.

```

1  template <class ShapeOptimizationProblem_SubClass>
2  TimeDependence<
3      ShapeOptimizationProblem_SubClass
4  > & TimeDependence<
5      ShapeOptimizationProblem_SubClass
6  >::operator=(const TimeDependenceType::Type & type_enum) {
7      M_type = type_enum;
8
9      // Setup cost functional time dependence
10     if (type_enum == TimeDependenceType::unsteady) {
11         M_timeData = TimeData(fake_this->shapeOptimizationData().dataFile, fake_this->
12             shapeOptimizationData().section_timeDiscretization);
13     }
14     else { // steady: keep the default (null) TimeData
15         M_timeData = TimeData();
16     }
17
18     // Reserve space for state & adjoint solution
19     fake_this->stateSolutionPtr().reserve( 1 + M_timeData.endTime_StepNumber());
20     fake_this->adjointSolutionPtr().reserve( 1 + M_timeData.endTime_StepNumber());
21     return *this;
22 }

```

### A.3.4 Classe `ShapeOptimizationProblem<ShapeParametrization>`

La dichiarazione della classe `ShapeOptimizationProblem<ShapeParametrization>` viene suddivisa nelle seguenti sezioni, rispetto alle astrazioni (a)-(e) enunciate in precedenza

#### Costruttori e dati del problema

```

1  public:
2      ShapeOptimizationProblem(ShapeOptimizationData shapeOptimizationData);
3      virtual ~ShapeOptimizationProblem();
4      virtual const ShapeOptimizationData & shapeOptimizationData() const;
5  protected:
6      ShapeOptimizationProblem();
7      virtual bool verbose() const;
8      virtual bool isSimulationTypeOptimize() const;

```

```

9     virtual TimeDependence<ShapeOptimizationProblem<ShapeParametrization> > &
        timeDependence();
10 private:
11     const bool M_isSimulationTypeOptimize; // and not startup?
12     const ShapeOptimizationData M_shapeOptimizationData;
13     TimeDependence<ShapeOptimizationProblem<ShapeParametrization> > M_timeDependence;
14     friend class TimeDependence<ShapeOptimizationProblem<ShapeParametrization> >;

```

Il costruttore copia i dati del problema in `M_shapeOptimizationData` ed inizializza a valori di default gli altri attributi privati che verranno mostrati in seguito.

La classe espone in particolare il metodo `timeDependence()`, che sarà necessario chiamare nel costruttore della classe concreta in modo da specificare la dipendenza temporale del problema: grazie alla struttura dati presentata nella Appendice A.3.3 per impostare, ad esempio, un problema non stazionario basterà scrivere:

```

1 this->timeDependence() = ShapeOptimizationProblem::TimeDependenceType::unsteady;

```

### Astrazione (a) - Parametrizzazione della forma

```

1 public:
2     virtual boost::shared_ptr<const ShapeParametrization> shapePtr() const;
3 protected:
4     virtual boost::shared_ptr<ShapeParametrization> & shapePtr();
5 private:
6     boost::shared_ptr<ShapeParametrization> M_shape_ptr;

```

La parametrizzazione della forma è governata da un oggetto `M_shape_ptr` del tipo `template ShapeParametrization`: esso è inizializzato tramite la lettura del file di dati *GetPot*, e verrà modificato durante le iterazioni del ciclo di ottimizzazione.

### Astrazione (b<sub>1</sub>) - Espressione del funzionale costo e del suo gradiente

A questo livello di astrazione non è presente nessun metodo per l'espressione del funzionale costo e del suo gradiente.

### Astrazione (b<sub>2</sub>) - Calcolo del funzionale costo e del suo gradiente

```

1 public:
2     virtual LifeV::Real computeCostFunctional() = 0;
3 protected:
4     template <class ShapeOptimizationProblem_SubClass>
5     Matrix_GenericSize computeVolumeIntegral_OverTime(const ElementaryIntegrator_Base<
        ShapeOptimizationProblem_SubClass> & elementaryIntegrator);
6     template <class ShapeOptimizationProblem_SubClass>
7     Matrix_GenericSize computeVolumeIntegral(const ElementaryIntegrator_Base<
        ShapeOptimizationProblem_SubClass> & elementaryIntegrator, LifeV::UInt
        timeStepNumber=0);

```

dove la classe `ElementaryIntegrator_Base<ShapeOptimizationProblem_SubClass>` è definita da

```

1 typedef Eigen::Matrix<LifeV::Real, Eigen::Dynamic, Eigen::Dynamic> Matrix_GenericSize;
2 template <class ShapeOptimizationProblem_SubClass>
3 class ElementaryIntegrator_Base {
4 public:
5     ElementaryIntegrator_Base(ShapeOptimizationProblem_SubClass* fake_this, LifeV::
        UInt rows, LifeV::UInt columns);
6     virtual Matrix_GenericSize operator()(LifeV::UInt idx_FESpace, const LifeV::
        CurrentFE& fe, const LifeV::DOF& dof, LifeV::UInt timeStepNumber=0) const = 0;
7     virtual LifeV::UInt derivativesNeeded(LifeV::UInt idx_FESpace) const = 0;
8     const LifeV::UInt & rows() const;

```

```

9     const LifeV::UInt & columns() const;
10  protected:
11     ShapeOptimizationProblem_SubClass* fake_this;
12  private:
13     LifeV::UInt M_rows;
14     LifeV::UInt M_columns;
15 };

```

Si tratta della classe astratta che calcola l'integrale su un singolo elemento di volume della mesh: tuttavia, siccome nel calcolo del funzionale costo e del suo gradiente si eseguono integrazioni delle soluzioni del problema di stato e del problema aggiunto, è necessario anche che esse vengano rese accessibili tramite il puntatore `fake_this`.

In particolare sia  $i = 1, \dots, N$ , dove  $N$  è il numero di spazi ad elementi finiti necessari per la discretizzazione dei problemi di stato ed aggiunto, e sia  $y_i$  una funzione appartenente all' $i$ -esimo spazio ad elementi finiti: l'assunzione fondamentale di questa classe è che, detta  $f$  l'integranda a valori matriciali, valga

$$\int_{\Omega} f(\mathbf{x}, t, y_1, y_2, \dots, y_N) d\mathbf{x} = \sum_{i=1}^N \int_{\Omega} f_i(\mathbf{x}, t, y_i) d\mathbf{x} \quad (\text{A.1})$$

per opportune funzioni  $f_i$ , in modo da evitare l'interpolazione tra diversi spazi ad elementi finiti.

Il metodo `Matrix_GenericSize operator() (LifeV::UInt idx_FESpace, const LifeV::CurrentFE& fe, const LifeV::DOF& dof, LifeV::UInt timeStepNumber)` esegue quindi

$$\int_v f_i(\mathbf{x}, t, y_i) d\mathbf{x}, \quad (\text{A.2})$$

essendo  $v = fe$ ,  $i = idx\_FESpace$ ,  $y_i$  calcolato a partire dai gradi di libertà contenuti in `dof` e  $t$  tempo associato all'iterazione temporale `timeStepNumber`.

La classe `ShapeOptimizationProblem` è quindi fornita di un metodo generale per eseguire l'integrazione sul dominio, `computeVolumeIntegral()`, il quale calcola la somma degli integrali su ciascun elemento di volume, eseguiti dall'`elementaryIntegrator` passato in `input`. Analogamente è possibile eseguire un'integrazione sia in tempo sia in spazio tramite `computeVolumeIntegral_OverTime()`, che approssima l'integrazione in tempo con una formula dei trapezi e si serve di `computeVolumeIntegral()` per ciascuna integrazione in spazio.

### Astrazione (c) - Risoluzione problemi di stato ed aggiunto

```

1  protected:
2     virtual void setupFESpaces() = 0;
3     virtual void solveStateEquation() = 0;
4     virtual void solveAdjointEquation() = 0;
5     virtual void solutionPostProcessing(LifeV::UInt iterationNumber) = 0;
6     virtual Solutions_VectorOverTime & stateSolutionPtr();
7     virtual Solutions_VectorOverTime & adjointSolutionPtr();
8     virtual typename FESpaces<typename ShapeParametrization::RegionMeshType>::Type &
     variablesFESpacesPtr();
9     virtual boost::shared_ptr<Epetra_Comm> & commPtr();
10 private:
11     Solutions_VectorOverTime M_stateSolution_ptr;
12     Solutions_VectorOverTime M_adjointSolution_ptr;
13     typename FESpaces<typename ShapeParametrization::RegionMeshType>::Type
     M_variablesFESpaces_ptr;
14     boost::shared_ptr<Epetra_Comm> M_comm_ptr;

```

I primi quattro metodi gestiscono la creazione degli spazi ad elementi finiti sulla mesh deformata, la risoluzione del problema di stato, la risoluzione del problema aggiunto e l'eventuale esportazione

dei risultati: è naturale che essi siano *pure virtual* in quanto verranno implementati solo al terzo livello di ereditarietà (quello del problema concreto). Gli altri quattro metodi garantiscono l'accesso alle variabili che memorizzano le soluzioni del problema di stato, le soluzioni del problema aggiunto, gli spazi ad elementi finiti (condivisi tra le corrispondenti soluzioni del problema di stato e di quello aggiunto), ed il partizionatore che è stato utilizzato nella suddivisione della mesh.

#### Astrazione (d) - Metodo di discesa

```

1 public:
2     virtual void run();
3 protected:
4     virtual void solve() = 0;

```

Il metodo più importante della classe è il metodo pubblico `run()`, che è l'unico metodo della successiva classe concreta che l'utente ha la necessità di chiamare per lanciare la simulazione. Questo metodo, dopo aver controllato se deve o meno lanciare la simulazione (attributo privato `M_isSimulationTypeOptimize`) chiama il metodo `solve()`, che a questo livello è *pure virtual* in quanto non è stato specificato quale sia il metodo di ottimizzazione da utilizzare.

#### Astrazione (e) - Vincoli

A questo livello di astrazione non è presente nessun metodo per la gestione dei vincoli sulla forma ottima.

### A.3.5 Classe `FFDShapeOptimizationProblem<RegionMesh>`

La dichiarazione della classe `FFDShapeOptimizationProblem<RegionMesh>` viene suddivisa nelle seguenti sezioni, rispetto alle astrazioni (a)-(e) enunciate in precedenza.

#### Costruttori ed Astrazione (a) - Parametrizzazione della forma

```

1 public:
2     FFDShapeOptimizationProblem(ShapeOptimizationData shapeOptimizationData);
3     virtual ~FFDShapeOptimizationProblem();
4     virtual const ShapeParametrization::VectorEnabledDisplacements_ptr &
        enabledDisplacements() const;
5     virtual const ShapeParametrization::VectorEnabledControlPoints_ptr &
        enabledControlPoints() const;
6     virtual void shapePreProcessing() = 0;
7 protected:
8     FFDShapeOptimizationProblem();
9     ShapeParametrization::VectorEnabledDisplacements_ptr M_enabledDisplacements;
10    ShapeParametrization::VectorEnabledControlPoints_ptr M_enabledControlPoints;

```

Il costruttore, oltre a richiamare quello della classe padre, inizializza gli attributi privati `M_enabledDisplacements`, vettore di puntatori agli spostamenti abilitati, e `M_enabledControlPoints`, vettore di puntatori ai punti di controllo con almeno uno spostamento abilitato<sup>2</sup>. È possibile eseguire alcune semplici operazioni di *preprocessing* della forma<sup>3</sup> mediante il metodo `shapePreProcessing()`.

<sup>2</sup>Seguendo la suddivisione introdotta nella Sezione 5.3, questi vettori contengono i gradi di libertà di *ottimizzazione*; ulteriori gradi di libertà di *deformazione* possono essere aggiunti mediante il metodo di *preprocessing*. Non è però possibile cambiare alcune proprietà della parametrizzazione nel corso del ciclo di ottimizzazione, in quanto questo cambiamento non si ripercuoterebbe su `M_enabledDisplacements` e `M_enabledControlPoints`. Ad esempio non è possibile, tra un'iterazione e la successiva, decidere di abilitare un ulteriore grado di libertà di ottimizzazione: questa limitazione tuttavia non è eccessivamente stringente dato che una scelta adattativa ed iterativa dei punti di controllo esula dagli scopi di questo lavoro.

<sup>3</sup>Ad esempio, se è noto a priori che, tra tutte le forme ottime, ne esiste almeno una che gode di una proprietà di simmetria rispetto ad un piano  $\sigma$ , è possibile effettuare tre scelte:

Astrazione ( $b_1$ ) - Espressione del funzionale costo e del suo gradiente

```

1  protected:
2      virtual LifeV::UInt costFunctionalDerivativesNeeded(LifeV::UInt idx_stateVariable)
          const = 0;
3      virtual LifeV::UInt costFunctionalGradientDerivativesNeeded(LifeV::UInt
          idx_stateAndAdjointVariable) const = 0;
4      virtual LifeV::Real costFunctionalExpression(LifeV::UInt idx_stateVariable,
          TimeIntegralType timeIntegralType, IntegratorData integratorData, std::vector<
          LifeV::Real> y, std::vector<Vector_SpaceDimensionSize> grad_y, std::vector<
          Matrix_SpaceDimensionSize> hess_y) = 0;
5      virtual Vector_SpaceDimensionSize
          costFunctionalGradientExpressionBernsteinComponent(LifeV::UInt
          idx_stateAndAdjointVariable, TimeIntegralType timeIntegralType, IntegratorData
          integratorData, std::vector<LifeV::Real> y, std::vector<LifeV::Real> p, std::
          vector<Vector_SpaceDimensionSize> grad_y, std::vector<
          Vector_SpaceDimensionSize> grad_p, std::vector<Matrix_SpaceDimensionSize>
          hess_y, std::vector<Matrix_SpaceDimensionSize> hess_p) = 0;
6      virtual LifeV::Real costFunctionalGradientExpressionGradBernsteinComponent(LifeV::
          UInt idx_stateAndAdjointVariable, TimeIntegralType timeIntegralType,
          IntegratorData integratorData, std::vector<LifeV::Real> y, std::vector<LifeV::
          Real> p, std::vector<Vector_SpaceDimensionSize> grad_y, std::vector<
          Vector_SpaceDimensionSize> grad_p, std::vector<Matrix_SpaceDimensionSize>
          hess_y, std::vector<Matrix_SpaceDimensionSize> hess_p) = 0;

```

dopo aver introdotto i seguenti tipi e le seguenti classi

```

1  typedef Eigen::Matrix<LifeV::Real, Eigen::Dynamic, 1> Vector_EnabledDisplacementsSize;
2  typedef Eigen::Matrix<LifeV::Real, Eigen::Dynamic, 1> Vector_SpaceDimensionSize;
3  typedef Eigen::Matrix<LifeV::Real, Eigen::Dynamic, Eigen::Dynamic>
          Matrix_SpaceDimensionSize;
4  class TimeIntegralType {
5  public:
6      enum Type {
7          Boundary = 0,
8          Interval
9      };
10     TimeIntegralType & operator=(const TimeIntegralType::Type & type_enum);
11     bool operator==(const TimeIntegralType::Type & type_enum) const;
12     bool operator!=(const TimeIntegralType::Type & type_enum) const;
13 private:
14     Type type;
15 };
16 struct IntegratorData {
17     ShapeParametrization::SimplePoint x;
18     LifeV::UInt timeStepNumber;
19     LifeV::Real t;
20     LifeV::ID meshElementID;
21     LifeV::UInt indexQuadraturePoint;
22 };

```

1. ottimizzare rispetto a gradi di libertà simmetrici rispetto al piano  $\sigma$ , senza però imporre alcun vincolo di simmetria: non è tuttavia garantito a priori che a convergenza si trovi una forma ottima simmetrica;
2. ottimizzare rispetto a gradi di libertà simmetrici rispetto al piano  $\sigma$ , imponendo il vincolo di simmetria: ciò comporta la necessità di inserire, oltre ai vincoli fisici presenti nel problema di ottimizzazione di forma, un ulteriore vincolo lineare per garantire la simmetria dei gradi di libertà;
3. ottimizzare rispetto a gradi di libertà da una sola parte del piano  $\sigma$ , ed imporre il valore ai gradi di libertà simmetrici implementando opportunamente il metodo `shapePreProcessing()`. Ciò comporta un risparmio di tempi computazionali, perché è necessario calcolare il gradiente rispetto ad un numero inferiore di spostamenti, ma a priori la forma che si ottiene a convergenza potrebbe dipendere dalla scelta di quale semispazio individuato da  $\sigma$  considerare soggetto ad ottimizzazione e quale aggiornato per simmetria.

Tutti i metodi e le classi sopra riportate sono necessarie per calcolare:

1. il funzionale costo:

◦ per un problema stazionario

$$J(\Omega) = \int_{\Omega} j(\Omega, \mathbf{x}, u_1, \dots, u_N) d\mathbf{x} = \sum_{i=1}^N \int_{\Omega} j_i(\Omega, \mathbf{x}, u_i) d\mathbf{x};$$

◦ per un problema non stazionario

$$J(\Omega) = \int_0^T \int_{\Omega} j(\Omega, \mathbf{x}, t, u_1, \dots, u_N) d\mathbf{x} dt + \int_{\Omega} j^T(\Omega, \mathbf{x}, T, u_1, \dots, u_N) d\mathbf{x} = \\ \sum_{i=1}^N \int_0^T \int_{\Omega} j_i(\Omega, \mathbf{x}, t, u_i) d\mathbf{x} dt + \sum_{i=1}^N \int_{\Omega} j_i^T(\mathbf{x}, T, \Omega, u_i) d\mathbf{x},$$

dove in entrambi i casi le variabili di stato sono indicate con  $u_i$  e l'ultima uguaglianza di ciascuna formula è dovuta a (A.1). Non è stato implementato alcun metodo per l'integrazione sul bordo del dominio, e quindi attualmente non è ammesso risolvere problemi in cui il funzionale costo dipenda da  $\partial\Omega$ .

2. gradiente del funzionale costo rispetto a  $\boldsymbol{\mu}_{l,k,m}$ , valutato in  $\boldsymbol{\mu}$ :

◦ per un problema stazionario esso è dato da (4.6). Imponendo nuovamente l'ipotesi (A.1) si giunge alla seguente espressione

$$\nabla_{\boldsymbol{\eta}_{k,l,m}} J(\Omega_{\boldsymbol{\mu}}) = \sum_{i=1}^N \int_{\Omega_{\boldsymbol{\mu}}} \mathfrak{D} dj_i^b(\Omega, \mathbf{x}_{\boldsymbol{\mu}}, u_i, p_i) b_{k,l,m}^{K,L,M}(\boldsymbol{\psi}(\mathbf{T}^{-1}(\mathbf{x}_{\boldsymbol{\mu}}; \boldsymbol{\mu}))) + \\ + \sum_{i=1}^N \int_{\Omega_{\boldsymbol{\mu}}} \mathfrak{D} \nabla \mathbf{T}^{-T}(\mathbf{x}_{\boldsymbol{\mu}}; \boldsymbol{\mu}) \mathfrak{D}^{-1} \nabla b_{k,l,m}^{K,L,M}(\boldsymbol{\psi}(\mathbf{T}^{-1}(\mathbf{x}_{\boldsymbol{\mu}}; \boldsymbol{\mu}))) dj_i^{\nabla b}(\Omega, \mathbf{x}_{\boldsymbol{\mu}}, u_i, p_i), \quad (\text{A.3})$$

dove  $u_i$  sono le variabili di stato e  $p_i$  le variabili aggiunte.

◦ per un problema non stazionario si può ricavare una formula analoga alla (A.3) in cui, come nel calcolo del funzionale costo, sia presente sia un integrale su  $[0, T] \times \Omega$ , sia un integrale su  $\{T\} \times \Omega$ .

Il metodo `costFunctionalExpression(LifeV::UInt idx_stateVariable, TimeIntegralType timeIntegralType, ...)` va implementato nella successiva classe concreta in modo che:

◦ in un problema stazionario

– se `timeIntegralType == Boundary`, restituisca

$$j_i(\Omega, \mathbf{x}, u_i(\mathbf{x}), \nabla u_i(\mathbf{x}), \nabla^2 u_i(\mathbf{x}));$$

– se `timeIntegralType == Interval`, restituisca un'eccezione;

◦ in un problema non stazionario

– se `timeIntegralType == Boundary`, restituisca

$$j_i^T(\Omega, \mathbf{x}, T, u_i(\mathbf{x}, T), \nabla u_i(\mathbf{x}, T), \nabla^2 u_i(\mathbf{x}, T));$$

– se `timeIntegralType == Interval`, restituisca

$$j_i(\Omega, \mathbf{x}, t, u_i(\mathbf{x}, t), \nabla u_i(\mathbf{x}, t), \nabla^2 u_i(\mathbf{x}, t)),$$

dove in ciascuno dei tre casi la dipendenza di  $j_i$  dalle derivate di  $u_i$  deve essere regolata da `costFunctionalDerivativesNeeded(LifeV::UInt idx_stateVariable)`.

Analogamente, il metodo `costFunctionalGradientExpressionBernsteinComponent()` formula l'espressione analitica di  $d\mathbf{j}_i^b$  ed il metodo `costFunctionalGradientExpressionGradBernsteinComponent()` l'espressione di  $d\mathbf{j}_i^{\nabla b}$ .

### Astrazione (b<sub>2</sub>) - Calcolo del funzionale costo e del suo gradiente

```

1  template <class RegionMesh>
2  public:
3      virtual LifeV::Real computeCostFunctional();
4  protected:
5      virtual Vector_EnabledDisplacementsSize computeCostFunctionalGradient();
6      virtual std::vector<std::vector<std::vector<ShapeParametrization::SimplePoint> > >
          & referenceDomain_FEQuadraturePoints();
7      virtual void setupReferenceDomain_FEQuadraturePoints(std::vector<std::vector<std::
          vector<ShapeParametrization::SimplePoint> > > & whereToStoreTheResult);
8  private:
9      class M_ElementaryIntegrator_costFunctional: public ElementaryIntegrator_Base<
          FFDShapeOptimizationProblem<RegionMesh> > {
10     public:
11         M_ElementaryIntegrator_costFunctional(FFDShapeOptimizationProblem<RegionMesh
          >* fake_this);
12         Matrix_GenericSize operator()(LifeV::UInt idx_stateVariable, const LifeV::
          CurrentFE& fe, const LifeV::DOF& dof, LifeV::UInt timeStepNumber=0)
          const;
13         LifeV::UInt derivativesNeeded(LifeV::UInt idx_stateVariable) const;
14         const TimeIntegralType & timeIntegralType() const;
15         TimeIntegralType & timeIntegralType();
16     private:
17         TimeIntegralType M_timeIntegralType;
18 };
19 friend class M_ElementaryIntegrator_costFunctional;
20 std::vector<std::vector<std::vector<ShapeParametrization::SimplePoint> > >
          M_referenceDomain_FEQuadraturePoints;
21 class M_ElementaryIntegrator_costFunctionalGradient:
22     public ElementaryIntegrator_Base<FFDShapeOptimizationProblem<RegionMesh> > {
23     public:
24         M_ElementaryIntegrator_costFunctionalGradient(FFDShapeOptimizationProblem<
          RegionMesh>* fake_this, ShapeParametrization::ControlPointID
          controlPointID);
25         Matrix_GenericSize operator()(LifeV::UInt idx_stateAndAdjointVariable, const
          LifeV::CurrentFE& fe, const LifeV::DOF& dof, LifeV::UInt timeStepNumber
          =0) const;
26         LifeV::UInt derivativesNeeded(LifeV::UInt idx_stateAndAdjointVariable) const
          ;
27         const TimeIntegralType & timeIntegralType() const;
28         TimeIntegralType & timeIntegralType();
29     private:
30         ShapeParametrization::ControlPointID controlPointID;
31         TimeIntegralType M_timeIntegralType;
32 };
33 friend class M_ElementaryIntegrator_costFunctionalGradient;
34 Vector_SpaceDimensionSize computeCostFunctionalGradient(ShapeParametrization::
          ControlPointID controlPointID);
35 };

```



È possibile eseguire il calcolo del funzionale costo e del suo gradiente derivando opportunamente due integratori elementari da `ElementaryIntegrator_Base` e sfruttando, per fornire l'espressione della funzione integranda, i metodi introdotti nell'astrazione ( $b_1$ ).

Il calcolo del funzionale costo è quindi eseguito dal seguente metodo:

```

1 template <class RegionMesh>
2 LifeV::Real FFDShapeOptimizationProblem<
3   RegionMesh
4 >::computeCostFunctional() {
5   M_ElementaryIntegrator_costFunctional elementaryIntegrator(this);
6   Matrix_GenericSize sum(
7     Matrix_GenericSize::Zero(elementaryIntegrator.rows(), elementaryIntegrator.
8       columns())
9   );
10  // 1. Compute the integral, over the domain and over time, of the cost functional
11  elementaryIntegrator.timeIntegralType() = TimeIntegralType::Interval;
12  sum += this->template computeVolumeIntegral_OverTime<FFDShapeOptimizationProblem<
13    RegionMesh> >(elementaryIntegrator);
14  // 2. Compute the integral, over the domain and at the final time, of the cost
15  //    functional
16  elementaryIntegrator.timeIntegralType() = TimeIntegralType::Boundary;
17  sum += this->template computeVolumeIntegral<FFDShapeOptimizationProblem<RegionMesh
18    > >(elementaryIntegrator, this->timeDependence().timeData().endTime_StepNumber
19    ());
20  // 3. Convert from generic matrix to Real
21  return sum(0,0);
22 }

```

Il calcolo del gradiente del funzionale costo rispetto agli spostamenti abilitati è eseguito dal seguente metodo:

```

1 template <class RegionMesh>
2 Vector_EnabledDisplacementsSize FFDShapeOptimizationProblem<
3   RegionMesh
4 >::computeCostFunctionalGradient() {
5   // vector containing the gradient w.r.t. the enabled displacements
6   Vector_EnabledDisplacementsSize gradientEnabledDisplacement(
7     Vector_EnabledDisplacementsSize::Zero(M_enabledDisplacements.size()));
8   LifeV::UInt iGradientEnabledDisplacement(0);
9   for(LifeV::UInt iEnabledCP(0); iEnabledCP < M_enabledControlPoints.size(); ++
10     iEnabledCP) {
11     if (this->verbose())
12       std::cout << "* Computing the cost functional gradient w.r.t to the enabled
13         displacements of control point" << *M_enabledControlPoints[iEnabledCP]
14         << std::endl;
15     // 1. compute the CostFunctionalGradient
16     Vector_SpaceDimensionSize gradientSingleCP(
17       computeCostFunctionalGradient(M_enabledControlPoints[iEnabledCP]->ID())
18     );
19     // 2. push gradientSingleCP on the bottom of gradientEnabledDisplacement,
20     // (only for the enabled displacements, of course you know that
21     // #enabled displacements >= 1)
22     for (LifeV::UInt iDisplacements(0); iDisplacements < M_enabledControlPoints[
23       iEnabledCP]->spaceDimension(); ++iDisplacements)
24       if (M_enabledControlPoints[iEnabledCP]->operator [] (iDisplacements).
25         isEnabled()) {
26         gradientEnabledDisplacement(iGradientEnabledDisplacement) =
27           gradientSingleCP[iDisplacements];
28         ++iGradientEnabledDisplacement;
29       }
30   }
31 }

```

```

26     return gradientEnabledDisplacement;
27 }

```

che si occupa di calcolare il gradiente rispetto a ciascun punto di controllo in `M_enabledControlPoints` (tramite la chiamata di riga 13 a `computeCostFunctionalGradient(ControlPointID controlPointID)`), che non viene riportato in quanto molto simile al metodo `computeCostFunctional()`, conservandone solo le componenti corrispondenti a spostamenti abilitati (riga 16-25).

### Astrazione (c) - Risoluzione problemi di stato ed aggiunto

Questa classe si limita ad ereditare i metodi della classe padre senza né aggiungerne altri, né fornire l'implementazione di eventuali metodi *pure virtual*.

### Astrazione (d) - Metodo di discesa

Questa classe si limita ad ereditare i metodi della classe padre senza né aggiungerne altri, né fornire l'implementazione di eventuali metodi *pure virtual*.

### Astrazione (e) - Vincoli

Questa classe si limita ad ereditare i metodi della classe padre senza né aggiungerne altri, né fornire l'implementazione di eventuali metodi *pure virtual*.

## A.3.6 Classe `FFDProjectionMethodShapeOptimizationProblem<RegionMesh>`

La dichiarazione della classe `FFDProjectionMethodShapeOptimizationProblem<RegionMesh>` viene suddivisa nelle seguenti sezioni, rispetto alle astrazioni (a)-(e) enunciate in precedenza.

### Costruttori

```

1 public:
2     FFDProjectionMethodShapeOptimizationProblem(ShapeOptimizationData
           shapeOptimizationData, NumericalOptimization::ProjectionMethodData
           optimizerData);
3     virtual ~FFDProjectionMethodShapeOptimizationProblem();
4 protected:
5     FFDProjectionMethodShapeOptimizationProblem();
6 private:
7     NumericalOptimization::ProjectionMethodData M_optimizerData;

```

Il costruttore, oltre a richiamare quello della classe padre, inizializza `M_optimizerData`, copiando in esso i dati necessari all'esecuzione del metodo del gradiente con proiezione.

### Astrazione (a) - Parametrizzazione della forma

La scelta della parametrizzazione della forma è completamente gestita nella classe padre, e quindi non c'è bisogno di introdurre ulteriori metodi.

### Astrazione (b<sub>1</sub>) - Espressione del funzionale costo e del suo gradiente

Questa classe si limita ad ereditare i metodi della classe padre senza né aggiungerne altri, né fornire l'implementazione di eventuali metodi *pure virtual*.

### Astrazione (b<sub>2</sub>) - Calcolo del funzionale costo e del suo gradiente

Il calcolo del funzionale costo e del suo gradiente rispetto alle variabili di ottimizzazione  $\mu$  è completamente gestito nella classe padre, e quindi non c'è bisogno di introdurre ulteriori metodi.

### Astrazione (c) - Risoluzione problemi di stato ed aggiunto

Questa classe si limita ad ereditare i metodi della classe padre senza né aggiungerne altri, né fornire l'implementazione di eventuali metodi *pure virtual*.

### Astrazione (d) - Metodo di discesa

I metodi che gestiscono la creazione e la configurazione di un oggetto `ProjectionMethod` sono i seguenti:

```

1 public:
2     virtual const NumericalOptimization::ProjectionMethodData & optimizerData() const;
3 protected:
4     class M_ProjectionMethod_Function: public NumericalOptimization::Function {
5     public:
6         M_ProjectionMethod_Function(FFDProjectionMethodShapeOptimizationProblem<
7             RegionMesh>* fake_this);
8         LifeV::Real operator() (const Vector_EnabledDisplacementsSize &
9             enabledDisplacementsValues);
10        Vector_EnabledDisplacementsSize gradient(const
11            Vector_EnabledDisplacementsSize &enabledDisplacementsValues);
12    private:
13        FFDProjectionMethodShapeOptimizationProblem<RegionMesh>* fake_this;
14    };
15 friend class M_ProjectionMethod_Function;
16 class M_ProjectionMethod_Exporter: public NumericalOptimization::Exporter {
17 public:
18     M_ProjectionMethod_Exporter(FFDProjectionMethodShapeOptimizationProblem<
19         RegionMesh>* fake_this);
20     void operator() (const NumericalOptimization::Result & min, const std::vector<
21         LifeV::Real> & backtrackingStepsizes, const std::vector<LifeV::Real> &
22         backtrackingIntermediateFunctionValues, const std::vector<LifeV::Real> &
23         stoppingCriteriaValues, const Vector_EnabledDisplacementsSize & gradient
24         , const LifeV::UInt & iterationNumber);
25 private:
26     FFDProjectionMethodShapeOptimizationProblem<RegionMesh>* fake_this;
27 };
28 friend class M_ProjectionMethod_Exporter;
29 virtual void solve();

```

Il metodo `solve()` crea un oggetto `ProjectionMethod` e, prima di lanciare l'ottimizzazione, imposta come funzione costo un'istanza di `M_ProjectionMethod_Function`: essa contiene l'`operator()`, responsabile di tutte le operazioni necessarie per la risoluzione dell'equazione di stato e per la valutazione del funzionale costo alla corrente iterazione (cioè fissati i valori di  $\mu$ , e quindi fissata la forma).

```

1 template <class RegionMesh>
2 LifeV::Real FFDProjectionMethodShapeOptimizationProblem<
3     RegionMesh
4 >::M_ProjectionMethod_Function::operator() (
5     const Vector_EnabledDisplacementsSize &enabledDisplacementsValues
6     ) {
7     // 0. Reset the solution vector
8     fake_this->stateSolutionPtr().clear();
9
10    // 1. Update the shape

```

```

11     for (LifeV::UInt d(0); d < fake_this->enabledDisplacements().size(); ++d)
12         fake_this->enabledDisplacements()[d]->value(enabledDisplacementsValues[d]);
13     fake_this->shapePreProcessing();
14     // Deform the domain
15     if (fake_this->verbose())
16         std::cout << "Deforming the mesh" << std::endl;
17     fake_this->shapePtr()->deformReferenceDomain();
18     try {
19         // check for negative volumes
20         fake_this->shapePtr()->checkDeformedDomain();
21     }
22     catch (ShapeParametrization::FFDBadMeshException e) {
23         // make sure NOT to use this mesh: return an "infinite" cost!
24         if (fake_this->verbose())
25             std::cout << "Mesh was corrupted by the deformation: report \"infinite\"
26                 cost" << std::endl;
27         return std::numeric_limits<LifeV::Real>::max();
28     }
29 #ifdef HAVE_MPI
30     MPI_Barrier(MPI_COMM_WORLD);
31 #endif
32     // 2. Setup FEM spaces
33     if (fake_this->verbose())
34         std::cout << "Setup of the FE spaces on the deformed mesh" << std::endl;
35     fake_this->setupFESpaces();
36 #ifdef HAVE_MPI
37     MPI_Barrier(MPI_COMM_WORLD);
38 #endif
39
40     // 3. Solve the state problem
41     if (fake_this->verbose())
42         std::cout << "Solving the state problem" << std::endl;
43     fake_this->solveStateEquation();
44 #ifdef HAVE_MPI
45     MPI_Barrier(MPI_COMM_WORLD);
46 #endif
47
48     // 4. Get the value of the cost functional
49     if (fake_this->verbose())
50         std::cout << "Computing the cost functional" << std::endl;
51     LifeV::Real cost(fake_this->computeCostFunctional());
52     if (fake_this->verbose())
53         std::cout << "* Cost Functional = " << std::setprecision(10) << cost << std::
54             endl;
55 #ifdef HAVE_MPI
56     MPI_Barrier(MPI_COMM_WORLD);
57 #endif
58     return cost;
59 }

```

Analogamente, sempre a forma fissata, il metodo `gradient()` esegue le operazioni necessarie per la risoluzione dell'equazione aggiunta e la valutazione del gradiente del funzionale costo.

```

1 template <class RegionMesh>
2 Vector_EnabledDisplacementsSize FFDProjectionMethodShapeOptimizationProblem<
3     RegionMesh
4 >::M_ProjectionMethod_Function::gradient(
5     const Vector_EnabledDisplacementsSize & /*enabledDisplacementsValues*/
6 ) {
7     // 0. Reset the solution vector
8     fake_this->adjointSolutionPtr().clear();
9

```

```

10 // 1. Update the shape
11 // Since the order in the optimization cycle is function evaluation and then
    function gradient evaluation, we don't need to update the shape as it was done
    in section 1. of function evaluation
12
13 // 2. Setup FEM spaces
14 // As in 1., FEM spaces are already up-to-date
15
16 // 3. Solve the adjoint problem
17 if (fake_this->verbose())
18     std::cout << "Solving the adjoint problem" << std::endl;
19     fake_this->solveAdjointEquation();
20 #ifdef HAVE_MPI
21     MPI_Barrier(MPI_COMM_WORLD);
22 #endif
23
24 // 4. Get the value of the cost functional gradient
25 if (fake_this->verbose())
26     std::cout << "Computing the cost functional gradient w.r.t the enabled
    displacements" << std::endl;
27     Vector_EnabledDisplacementsSize gradient(fake_this->computeCostFunctionalGradient
    ());
28 #ifdef HAVE_MPI
29     MPI_Barrier(MPI_COMM_WORLD);
30 #endif
31
32     return gradient;
33 }

```

Entrambi questi metodi hanno la seguente struttura comune:

1. deforma la mesh secondo gli spostamenti passati in ingresso, eseguendo l'eventuale *preprocessing* dei gradi di libertà (metodo `shapePreProcessing()` ) ed invocando successivamente il metodo `deformReferenceDomain()` della classe `FreeFormDeformation` per effettuare la deformazione;
2. esegui il *setup* degli spazi ad elementi finiti sulla mesh deformata;
3. risolvi il problema di stato (nel caso della valutazione del funzionale costo) o il problema aggiunto (nel caso della valutazione del gradiente del funzionale costo);
4. calcola il valore del funzionale costo (rispettivamente, del suo gradiente).

### Astrazione (e) - Vincoli

```

1 protected:
2     virtual bool & boundConstrained();
3     virtual LifeV::UInt & numberOfLinearConstraints();
4     virtual LifeV::UInt & numberOfNonlinearConstraints();
5     virtual NumericalOptimization::Bounds boundConstraints() = 0;
6     virtual NumericalOptimization::LinearConstraint linearConstraint(LifeV::ID
    idx_constraint) = 0;
7     virtual NumericalOptimization::NonlinearConstraint nonlinearConstraint(LifeV::ID
    idx_constraint) = 0;
8 private:
9     bool M_boundConstrained;
10    LifeV::UInt M_numberOfLinearConstraints;
11    LifeV::UInt M_numberOfNonlinearConstraints;

```

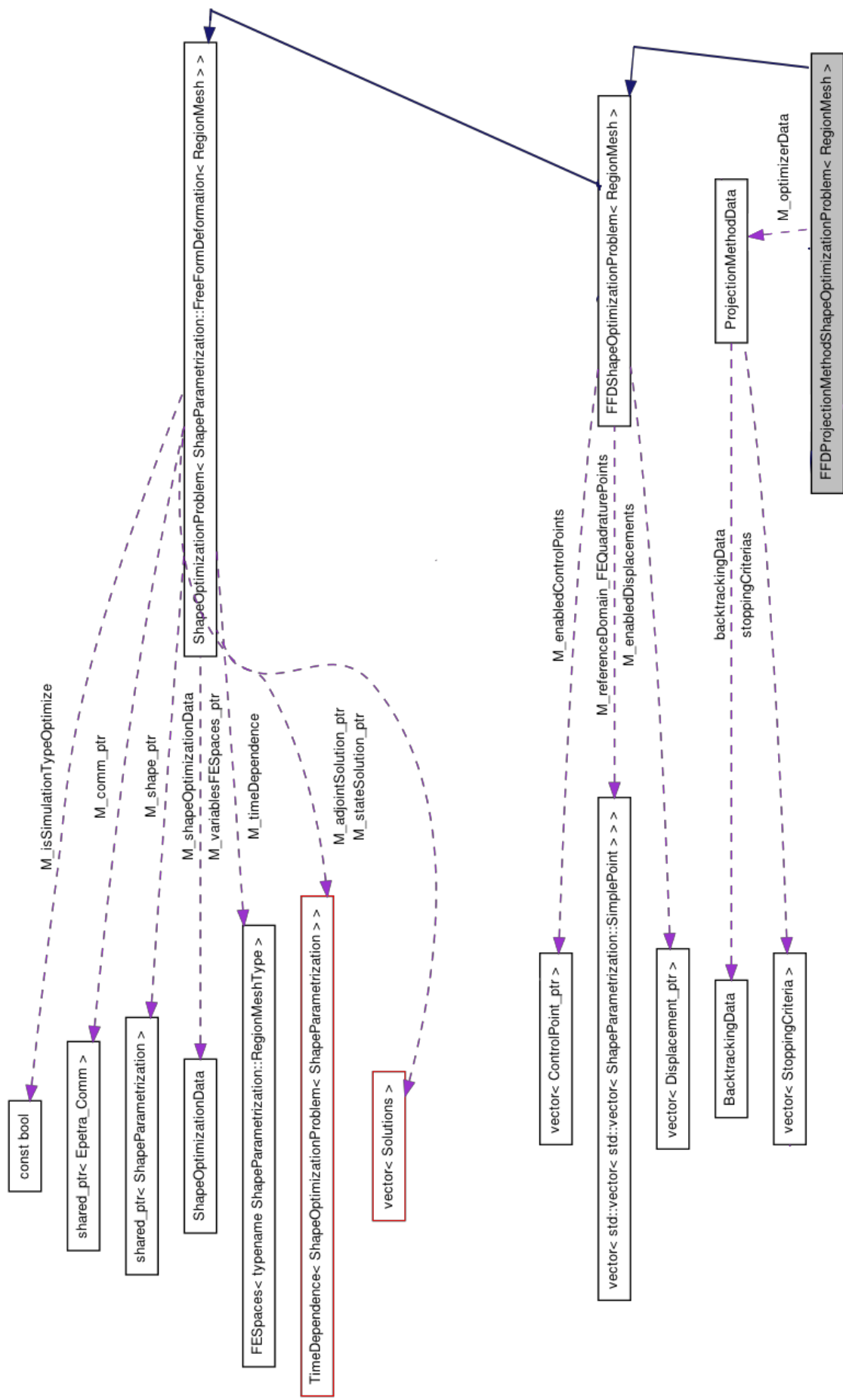


Figura A.3: Collaboration diagram della classe FFDPProjectionMethodShapeOptimizationProblem<RegionMesh>.

L'implementazione, nella successiva classe concreta, dei tre metodi *pure virtual* dovrà contenere la formulazione di eventuali vincoli sulla forma ottima (già formulati rispetto alle variabili  $\mu$ ).

### A.3.7 Classe `FFDProjectionMethodShapeOptimizationProblem_DecoratorBase<RegionMesh>`

Questa classe fornisce la base di un *decoratore* in grado di aggiungere vincoli al problema di ottimizzazione di forma. È necessario decorare solo due metodi: il metodo `solve()` (perché si occupa di leggere e caricare i vincoli nell'oggetto di tipo `ProjectionMethod`) ed il metodo `run()` (perché deve chiamare il metodo `solve()` decorato e non quello originale).

### A.3.8 Classe `AddVolumeConstraintDecorator<RegionMesh>`

Per imporre il vincolo di volume basta quindi sfruttare l'algoritmo di Newton-Raphson presentato nella Sezione 4.4.3 e decorare `nonlinearConstraint()` nel seguente modo:

```

1 template <class RegionMesh>
2 NumericalOptimization::NonlinearConstraint AddVolumeConstraintDecorator<
3   RegionMesh
4 >::nonlinearConstraint(LifeV::ID idx_constraint) {
5   if (idx_constraint == M_constraintID) {
6     NumericalOptimization::NonlinearConstraint volumeConstraint;
7     boost::shared_ptr<NumericalOptimization::Function> volumeConstraint_Function(
8       new M_volumeConstraintFunction(this));
9     volumeConstraint.constraintFunction = volumeConstraint_Function;
10    volumeConstraint.constraintType = M_constraintType;
11    return volumeConstraint;
12  } else {
13    return this->template FFDProjectionMethodShapeOptimizationProblem_DecoratorBase
14      <RegionMesh>::nonlinearConstraint(idx_constraint);
15  }
16 }
```

essendo `M_volumeConstraintFunction` la classe che si occupa della valutazione di  $V(\Omega_o(\mu))$  e  $\nabla_{\eta} V(\Omega_o(\mu))$ , effettuando le integrazioni tramite (A.3) con metodi analoghi a quelli forniti dalle classi precedenti.

## A.4 Un esempio: implementazione del caso test della Sezione 5.1.1

Il terzo livello di ereditarietà delle strutture dati introdotte in Appendice A.3 è rappresentato, per la risoluzione del problema di ottimizzazione di forma presentato nella Sezione 5.1.1, dalla classe `laplacianFFDProjectionMethodShapeOptimizationProblem<RegionMesh>`.

### Costruttori e dati del problema

Questa classe è dotata di un solo costruttore, il cui unico compito è quello di impostare la dipendenza temporale del problema:

```

1 this->timeDependence() = ShapeOptimizationProblem::TimeDependenceType::steady;
```

### Astrazione ( $b_1$ ) - Espressione del funzionale costo e del suo gradiente

Vengono implementati i metodi precedentemente lasciati *pure virtual*, in modo da specificare:

- o l'espressione del funzionale costo

```

1 // costFunctionalExpression(..)
2 if (idx_stateVariable == 0 && timeIntegralType == ShapeOptimizationProblem::
    TimeIntegralType::Boundary)
3     return pow(y[0]-optimalSolution(idx_stateVariable, integratorData.x),2);
4 else
5     // .. error ..

```

o l'espressione delle due componenti necessarie al calcolo del gradiente del funzionale costo

```

1 // costFunctionalGradientExpressionBernsteinComponent(..)
2 if (idx_stateAndAdjointVariable == 0 && timeIntegralType ==
    ShapeOptimizationProblem::TimeIntegralType::Boundary)
3     return 2*(y[0]-optimalSolution(idx_stateAndAdjointVariable, integratorData.x)
    )*(grad_y[0]-optimalSolutionGradient(idx_stateAndAdjointVariable,
    integratorData.x)) - (hess_y[0] - dirichletBCHessian(
    idx_stateAndAdjointVariable, integratorData.x))*grad_p[0] - hess_p[0]*(
    grad_y[0] - dirichletBCGradient(idx_stateAndAdjointVariable,
    integratorData.x));
4 else
5     // .. error ..

1 // costFunctionalGradientExpressionGradBernsteinComponent(..)
2 if (idx_stateAndAdjointVariable == 0 && timeIntegralType ==
    ShapeOptimizationProblem::TimeIntegralType::Boundary)
3     return -grad_p[0].dot(grad_y[0] - dirichletBCGradient(
    idx_stateAndAdjointVariable, integratorData.x)) + pow(y[0]-
    optimalSolution(idx_stateAndAdjointVariable, integratorData.x),2);
4 else
5     // .. error ..
6 }

```

### Astrazione (c) - Risoluzione problemi di stato ed aggiunto

Vengono implementati i quattro metodi *pure virtual* per la gestione degli spazi ad elementi finiti, la risoluzione dei problema di stato ed aggiunto e l'esportazione dei risultati.

La risoluzione dei problemi di stato ed aggiunto è effettuata grazie agli oggetti messi a disposizione da *LifeV*: in particolare *BCHandler* per la gestione delle condizioni al contorno, *ADRAssembler* per l'assemblamento del sistema ad elementi finiti e *SolverAztecOO* per la sua risoluzione.

### Astrazione (e) - Vincoli

L'istanza base della classe non contiene vincoli. Come indicato nella Sezione 5.1.1, in alcune simulazioni è stato aggiunto un vincolo di volume tramite il decoratore fornito dalla classe *AddVolumeConstraintDecorator*.

### Il main

Grazie alla struttura dati definita in precedenza, il *main* associato a questi casi test definisce solamente la creazione dell'oggetto contenente il problema concreto, una sua eventuale decorazione con vincoli aggiuntivi ed l'esecuzione della simulazione con il metodo *run()* :

```

1 // unconstrained problem
2 ShapeOptimizationData data(dataFileName);
3 ProjectionMethodData optData(dataFileName);
4 boost::shared_ptr<FFDProjectionMethodShapeOptimizationProblem<RegionMesh> >
    unconstrainedProblem(new laplacianFFDProjectionMethodShapeOptimizationProblem<
    RegionMesh>(data, optData));
5 unconstrainedProblem->run();

```



Numero di processori	Tempo di calcolo (s)
1	Memoria insufficiente
2	13091
4	5952
8	2886
16	1352
32	779
64	425
128	379
256	367

Tabella A.1: Analisi di scalabilità.

```

6 // constrained problem
7 AddVolumeConstraintDecorator_Data volumeConstraintData(dataFileName);
8 boost::shared_ptr<FFDProjectionMethodShapeOptimizationProblem<RegionMesh> >
  constrainedProblem(new AddVolumeConstraintDecorator<RegionMesh>(
  unconstrainedProblem, volumeConstraintData));
9 constrainedProblem->run();

```

### Analisi di scalabilità

Grazie all'integrazione con la versione parallela della libreria *LifeV*, è possibile variare il numero dei processori impiegati nella simulazione del problema di ottimizzazione di forma, aumentandoli in funzione della complessità del problema di stato, della dimensione della mesh, del numero dei gradi di libertà di ottimizzazione, ... Tuttavia, come ben noto nell'ambito del *calcolo parallelo*, l'aumento della potenza di calcolo non porta necessariamente ad un miglioramento delle prestazioni: per questo motivo è opportuno effettuare una *analisi di scalabilità* al variare del numero dei processori impiegati nella simulazione. Tale analisi è stata condotta, per la configurazione indicata con 4 nella Tabella 5.1, lanciando più volte l'eseguibile presentato in questa appendice sul cluster *Lagrange*<sup>4</sup>, per un numero di processori compreso tra 1 e 256. Come mostrato nella Tabella A.1, la libreria implementata in questa Tesi di Laurea Magistrale mostra una buona *performance* in ambito parallelo: il tempo di calcolo diminuisce all'aumentare del numero dei processori. Il miglioramento, inizialmente molto marcato per un basso numero di processori, si riduce sensibilmente a 128 processori: questo comportamento, molto comune nel calcolo parallelo, è indice del fatto che il tempo di calcolo complessivo è dominato dai tempi di comunicazione tra i processi paralleli, e dunque anche aumentando ulteriormente la potenza di calcolo non è possibile migliorare sensibilmente le prestazioni.

<sup>4</sup>208 nodi, ciascuno con due CPU Intel® Xeon® QuadCore, 3.166 GHz per nodo, con 16 GB di RAM per nodo. Cluster di calcolo offerto dal Consorzio Interuniversitario Lombardo per l'Elaborazione Automatica (CILEA).

# Ringraziamenti

Giunto al termine di questa Tesi, desidero ringraziare tutte le persone che mi hanno aiutato a raggiungere questo importante traguardo ed hanno contribuito allo sviluppo di questo lavoro.

Anzitutto, il primo e più importante ringraziamento è rivolto ai miei relatori, che mi hanno seguito e incoraggiato con passione ed enorme disponibilità: grazie al Prof. Salsa per tutte le proficue discussioni e per essere stato il punto di riferimento a cui rivolgermi con le domande più disparate; grazie ad Andrea per avermi introdotto – ormai un anno e mezzo fa! – ai problemi di ottimizzazione di forma, e per avermi guidato – da quel momento fino ad oggi – in tutte le fasi di questo lavoro, dai primi consigli sui riferimenti bibliografici fino alle ultime, dettagliatissime, correzioni di questo elaborato.

Ringrazio inoltre Jean Bonnemain, Antonio Cervone, Simone Deparis, Luca Formaggia, Alessio Fumagalli, Luca Paglieri e Matteo Pozzoli per l'importante contributo in alcuni degli aspetti computazionali di questa Tesi.

Desidero anche ringraziare i compagni di corso che hanno condiviso con me il percorso di questi anni, in particolare Fabio, Francesco, Matteo, Roberto e Simone.

L'ultimo, e più grande, "grazie" è rivolto alla mia famiglia, che mi ha sempre supportato con affetto e pazienza in questi anni.

# Bibliografia

- [ABH05] F. Abraham, M. Behr, e M. Heinkenschloss, *Shape optimization in stationary blood flow: a numerical study of non-Newtonian effects*, Computer Methods in Biomechanics and Biomedical Engineering, vol. 8, 2005, pp. 127–137.
- [All07] G. Allaire, *Conception optimale de structures*, Springer-Verlag Berlin-Heidelberg, 2007.
- [AN08] E. Amoiralis e I. Nikolos, *Freeform deformation versus B-spline representation in inverse airfoil design*, Journal of Computing and Information Science in Engineering **8** (2008), no. 2, 1–13.
- [AP06] G. Allaire e O. Pantz, *Structural optimization with FreeFem++*, Structural and Multidisciplinary Optimization **32** (2006), 173–181.
- [AQR07] V. Agoshkov, A. Quarteroni, e G. Rozza, *Shape design in aorto-coronary bypass anastomoses using perturbation theory*, SIAM journal on numerical analysis **44** (2007), no. 1, 367–384.
- [BB05] D. Bucur e G. Buttazzo, *Variational methods in shape optimization problems*, Progress in Nonlinear Differential Equations and Their Applications, Birkhäuser Boston, 2005.
- [BD00] C. Bertolotti e V. Deplano, *Three-dimensional numerical simulations of flow through a stenosed coronary bypass*, Journal of Biomechanics **33** (2000), no. 8, 1011–1022.
- [BDFD01] C. Bertolotti, V. Deplano, J. Fusseri, e P. Dupouy, *Numerical and experimental models of post-operative realistic flows in stenosed coronary bypasses*, Journal of Biomechanics **34** (2001), no. 8, 1049–1064.
- [Ber76] D. Bertsekas, *On the Goldstein-Levitin-Polyak gradient projection method*, Automatic Control, IEEE Transactions on **21** (1976), no. 2, 174–184.
- [Ber98] M. Berggren, *Numerical solution of a flow-control problem: Vorticity reduction by dynamic boundary action*, SIAM Journal on Scientific Computing **19** (1998), 829–860.
- [BFCLS97] J. Bello, E. Fernández-Cara, J. Lemoine, e J. Simon, *The differentiability of the drag with respect to the variations of a Lipschitz domain in a Navier–Stokes flow*, SIAM Journal on Control and Optimization **35** (1997), 626–640.
- [Bou74] J. M. Bourot, *On the numerical computation of the optimum profile in Stokes flow*, Journal of Fluid Mechanics **65** (1974), no. 3, 513–515.
- [Céa86] J. Céa, *Conception optimale ou identification de formes, calcul rapide de la dérivée directionnelle de la fonction coût*, Math. Model. Num. Anal **20** (1986), no. 3, 371–402.

- [Coq90] S. Coquillart, *Extended free-form deformation: a sculpturing tool for 3D geometric modeling*, International Conference on Computer Graphics and Interactive Techniques: Proceedings of the 17th annual conference on Computer graphics and interactive techniques, vol. 24, 1990, pp. 187–196.
- [DBB01] V. Deplano, C. Bertolotti, e O. Boiron, *Numerical simulations of unsteady flows in a stenosed coronary bypass graft*, Medical and Biological Engineering and Computing **39** (2001), no. 4, 488–499.
- [DOYM11] H. Do, A. Owida, W. Yang, e Y. Morsi, *Numerical simulation of the haemodynamics in end-to-side anastomoses*, International Journal for Numerical Methods in Fluids **67** (2011), 638–650.
- [DZ01] M. C. Delfour e J. P. Zolésio, *Shapes and geometries: analysis, differential calculus, and optimization*, SIAM, Philadelphia, 2001.
- [DZ08] C. D’Angelo e P. Zunino, *A numerical study of the interaction of blood flow and drug release from cardiovascular stents*, Numerical Mathematics and Advanced Applications. Proceedings of ENUMATH 2007, the 7th European Conference on Numerical Mathematics and Advanced Applications, Graz, Austria, September 2007 (K. Kunisch, G. Of, e O. Steinbach, eds.), Springer Berlin-Heidelberg, 2008, pp. 75–82.
- [Eva98] L. C. Evans, *Partial differential equations*, American Mathematical Society, 1998.
- [FQV09] L. Formaggia, A. Quarteroni, e A. Veneziani, *Cardiovascular mathematics: modeling and simulation of the circulatory system*, MS&A series, vol. 1, Springer-Verlag Italia, 2009.
- [FTR94] D. Y. Fei, J. D. Thomas, e S. E. Rittgers, *The effect of angle and flow rate upon hemodynamics in distal vascular graft anastomoses: a numerical model study*, J. Biomech. Eng. **116** (1994), no. 3, 331–336.
- [Fuj88] N. Fujii, *Lower semicontinuity in domain optimization problems*, Journal of optimization theory and applications **59** (1988), no. 3, 407–422.
- [GK98] M. D. Gunzburger e H. Kim, *Existence of an optimal solution of a shape control problem for the stationary Navier–Stokes equations*, SIAM J. Control Optim. **36** (1998), 895–909.
- [GMZ08a] Z. Gao, Y. Ma, e H. Zhuang, *Optimal shape design for the time-dependent Navier–Stokes flow*, International Journal for Numerical Methods in Fluids **57** (2008), no. 10, 1505–1526.
- [GMZ08b] ———, *Shape optimization for Stokes flow*, Applied Numerical Mathematics **58** (2008), no. 6, 827–844.
- [Gol64] A. A. Goldstein, *Convex programming in Hilbert space*, Bulletin of The American Mathematical Society **70** (1964), 709–711.
- [GR86] V. Girault e P. Raviart, *Finite element methods for Navier–Stokes equations: theory and algorithm*, Springer series in computational mathematics, Springer-Verlag Berlin-Heidelberg, 1986.
- [Gun03] M. D. Gunzburger, *Perspectives in flow control and optimization*, SIAM, Philadelphia, 2003.

- [HCHH96] F. S. Henry, M. W. Collins, P. E. Hughes, e T. V. How, *Numerical investigation of steady flow in proximal and distal end-to-side anastomoses*, J. Biomech. Eng. **118** (1996), no. 3, 302–310.
- [Hem73] W. Hemp, *Optimum structures*, Clarendon Press, Oxford, 1973.
- [HH78] R. M. Hicks e P. A. Henne, *Wing design by numerical optimization*, Journal of Aircraft **15** (1978), no. 7, 407–412.
- [HM03] J. Haslinger e R. A. E. Mäkinen, *Introduction to shape optimization: theory, approximation, and computation*, SIAM, Philadelphia, 2003.
- [HMV74] R. M. Hicks, E. M. Murman, e G. N. Vanderplaats, *An assessment of airfoil design by numerical optimization*, NASA TM X-3092, Ames Research Center, Moffet Field, California, 1974.
- [HNF<sup>+</sup>99] D. W. Holdsworth, C. J. D. Norley, R. Frayne, D. A. Steinman, e B. K. Rutt, *Characterization of common carotid artery blood-flow waveforms in normal human subjects*, Physiological measurement **20** (1999), 219–240.
- [HOWF09] J. Hurst, R. O'Rourke, R. Walsh, e V. Fuster, *Hurst's the heart manual of cardiology*, McGraw-Hill Medical, 2009.
- [HP05] A. Henrot e M. Pierre, *Variation et optimisation de formes: une analyse géométrique*, Springer-Verlag Berlin-Heidelberg, Paris, 2005.
- [HP10] A. Henrot e Y. Privat, *What is the optimal shape of a pipe?*, Archive for rational mechanics and analysis **196** (2010), no. 1, 281–302.
- [IMP96] F. Inzoli, F. Migliavacca, e G. Pennati, *Numerical analysis of steady flow in aorto-coronary bypass 3-D model*, J. Biomech. Eng. **118** (1996), no. 2, 172–179.
- [Jam88] A. Jameson, *Aerodynamic design via control theory*, Journal of Scientific Computing **3** (1988), no. 3, 233–260.
- [Jam95] ———, *Optimum aerodynamic design using CFD and control theory*, AIAA paper 95-1729, 12th AIAA Computational Fluid Dynamics Conference 1995 (1995), 926–949.
- [KGZG85] D. N. Ku, D. P. Giddens, C. K. Zarins, e S. Glagov, *Pulsatile flow and atherosclerosis in the human carotid bifurcation. Positive correlation between plaque location and low oscillating shear stress*, Arteriosclerosis, Thrombosis, and Vascular Biology **5** (1985), no. 3, 293–302.
- [Kir81] U. Kirsch, *Optimum structural design: concepts, methods, and applications*, McGraw-Hill, 1981.
- [KPT<sup>+</sup>00] B. Kawohl, O. Pironneau, L. Tartar, J. Zolésio, A. Cellina, e A. Ornelas, *Optimal shape design: lectures given at the joint C.I.M./C.I.M.E. summer school held in Tróia, Portugal, June 1-6, 1998*, Lecture notes in mathematics, Springer-Verlag Berlin Heidelberg, 2000.
- [Ku97] D. N. Ku, *Blood flow in arteries*, Annual Review of Fluid Mechanics **29** (1997), no. 1, 399–434.
- [LAK97] M. Lei, J. Archie, e C. Kleinstreuer, *Computational design of a bypass graft that minimizes wall shear stress gradients in the region of the distal anastomosis*, Journal of vascular surgery **25** (1997), no. 4, 637–646.

- [LFB08] F. Loth, P. Fischer, e H. Bassiouny, *Blood flow in end-to-side anastomoses*, *Annu. Rev. Fluid Mech.* **40** (2008), 367–393.
- [Lio71] J. Lions, *Optimal control of systems governed by partial differential equations*, vol. 170, Springer-Verlag Berlin Heidelberg, 1971.
- [LP66] E. Levitin e B. Polyak, *Constrained minimization problems*, *USSR. Comput. Math. Math. Phys.* **6** (1966), 1–50.
- [LPQR11] M. Lombardi, N. Parolini, A. Quarteroni, e G. Rozza, *Numerical simulation of sailing boats: dynamics, FSI, and shape optimization*, Report MOX 13/2011, MOX, Dipartimento di Matematica, Politecnico di Milano, 2011.
- [LR10] T. Lassila e G. Rozza, *Parametric free-form shape design with PDE models and reduced basis method*, *Computer Methods in Applied Mechanics and Engineering* **199** (2010), no. 23-24, 1583–1592.
- [Man08] A. Manzoni, *Ottimizzazione di forma per problemi di fluidodinamica: analisi teorica e metodi numerici*, Master’s thesis, MOX, Dipartimento di matematica, Politecnico di Milano, 2008.
- [Mar86] J. Marti, *Introduction to Sobolev spaces and finite element solution of elliptic boundary value problems*, Academic Press, London, 1986.
- [MFT08] A. Marsden, J. Feinstein, e C. Taylor, *A computational framework for derivative-free optimization of cardiovascular geometries*, *Computer Methods in Applied Mechanics and Engineering* **197** (2008), no. 21-24, 1890–1905.
- [MGP<sup>+</sup>07] F. Migliavacca, F. Gervaso, M. Prosi, P. Zunino, S. Minisini, L. Formaggia, e G. Dubini, *Expansion and drug elution model of a coronary stent*, *Computer Methods in Biomechanics and Biomedical Engineering* **10** (2007), no. 1, 63–73.
- [MNPV11] P. Morin, R. Nocketto, S. Pauletti, e M. Verani, *AFEM for shape optimization*, Report MOX 29/2011, MOX, Dipartimento di Matematica, Politecnico di Milano, 2011.
- [MP10] B. Mohammadi e O. Pironneau, *Applied shape optimization for fluids*, *Numerical Mathematics and Scientific Computation*, Oxford Univ. Press, New York, NY, 2010.
- [MQR11] A. Manzoni, A. Quarteroni, e G. Rozza, *Shape optimization for viscous flows by reduced basis methods and free-form deformation*, *Int. J. Numer. Meth. Fluids*, In press, 2011.
- [MS76] F. Murat e J. Simon, *Sur le contrôle par un domaine géométrique*, Internal report no 76 015, Laboratoire d’Analyse Numérique de l’Université Paris 6, 1976.
- [NW99] J. Nocedal e S. Wright, *Numerical optimization*, Springer-Verlag New York, 1999.
- [OK03] Y. Ogawa e M. Kawahara, *Shape optimization of body located in incompressible viscous flow based on optimal control theory*, *International Journal of Computational Fluid Dynamics* **17** (2003), 243–251.
- [PF08] P. Pavone e M. Fioranelli, *Malattia coronarica: fisiopatologia e diagnostica non invasiva con TC*, Springer-Verlag Italia, 2008.
- [Pir73] O. Pironneau, *On optimum profiles in Stokes flow*, *Journal of Fluid Mechanics* **59** (1973), no. 1, 117–128.

- [Pir84] ———, *Optimal shape design for elliptic systems*, Springer series in computational physics, Springer-Verlag New York, 1984.
- [PLN<sup>+</sup>10] M. Probst, M. Lülfsmann, M. Nicolai, H. Bücker, M. Behr, e C. Bischof, *Sensitivity of optimal shapes of artificial grafts with respect to flow parameters*, Computer methods in applied mechanics and engineering **199** (2010), no. 17-20, 997–1005.
- [PQ05] N. Parolini e A. Quarteroni, *Mathematical models and numerical simulations for the America's Cup*, Computer Methods in Applied Mechanics and Engineering **194** (2005), no. 9-11, 1001–1026.
- [Pra74] W. Prager, *Introduction to structural optimization*, CISM Lecture Notes, no. 212, Springer-Verlag, 1974.
- [PSC<sup>+</sup>08] A. K. Politis, G. P. Stavropoulos, M. N. Christolis, P. G. Panagopoulos, N. S. Vlachos, e N. C. Markatos, *Numerical modelling of simulated blood flow in idealized composite arterial coronary grafts: transient flow*, Journal of biomechanics **41** (2008), no. 1, 25–39.
- [QR03] A. Quarteroni e G. Rozza, *Optimal control and shape optimization of aorto-coronary bypass anastomoses*, Mathematical Models and Methods in Applied Sciences **13** (2003), no. 12, 1801–1824.
- [QTV00] A. Quarteroni, M. Tuveri, e A. Veneziani, *Computational vascular fluid dynamics: problems, models and methods*, Computing and Visualization in Science **2** (2000), no. 4, 163–197.
- [Qua08] A. Quarteroni, *Modellistica numerica per problemi differenziali*, Springer-Verlag Italia, 2008.
- [QV08] A. Quarteroni e A. Valli, *Numerical approximation of partial differential equations*, vol. 23, Springer-Verlag Berlin Heidelberg, 2008.
- [RBK95] G. I. N. Rozvany, M. P. Bendsøe, e U. Kirsch, *Layout Optimization of Structures*, Applied Mechanics Reviews **48** (1995), no. 2, 41–119.
- [Ric95] S. Richardson, *Optimum profiles in two-dimensional Stokes flow*, Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences **450** (1995), no. 1940, 603–622.
- [Roz89] G. Rozvany, *Structural design via optimality criteria: the Prager approach to structural optimization*, Kluwer Academic, 1989.
- [Roz05a] G. Rozza, *On optimization, control and shape design of an arterial bypass*, International Journal for Numerical Methods in Fluids **47** (2005), no. 10-11, 1411–1419.
- [Roz05b] ———, *Shape design by optimal flow control and reduced basis techniques: applications to bypass configurations in haemodynamics*, Ph.D. thesis, École Polytechnique Fédérale de Lausanne, 2005.
- [RP76] T. Ruch e H. Patton, *Fisiologia e biofisica di J.F. Fulton-W.H. Howell*, Società Editrice Universo, Roma, 1976.
- [Sal08] S. Salsa, *Partial Differential Equations in Action - From Modelling to Theory*, Springer-Verlag Italia, 2008.

- [SCGT05] M. Sankaranarayanan, L. Chua, D. Ghista, e Y. Tan, *Computational model of blood flow in the aorto-coronary bypass graft*, Biomedical engineering online **4** (2005), no. 1, 14–27.
- [SM10] S. Sankaran e A. L. Marsden, *The impact of uncertainty on shape optimization of idealized bypass graft models in unsteady flow*, Physics of Fluids **22** (2010), no. 12, 1–16.
- [SP86] T. Sederberg e S. Parry, *Free-form deformation of solid geometric models*, International Conference on Computer Graphics and Interactive Techniques: Proceedings of the 13th annual conference on Computer graphics and interactive techniques, vol. 20, 1986, pp. 151–159.
- [Sve93] V. Sverák, *On optimal shape design*, J. Math. Pures Appl **72** (1993), no. 6, 537–551.
- [SZ92] J. Sokolowski e J. P. Zolésio, *Introduction to shape optimization: Shape Sensitivity Analysis*, Springer-Verlag Berlin-Heidelberg, 1992.
- [TD04] C. Taylor e M. Draney, *Experimental and computational methods in cardiovascular fluid mechanics*, Annu. Rev. Fluid Mech. **36** (2004), 197–231.
- [Tem79] R. Temam, *Navier-Stokes equations: theory and numerical analysis*, North-Holland, 1979.
- [VJB11] J. Vimmr, A. Jonášová, e O. Bublík, *Pulsatile blood flow modelling in realistic aorto-coronary bypass models*, Proceedings of the 4th international conference on Modelling of mechanical and mechatronic systems 2011, Herl’any, Slovakia, 20.-22.9.2011, 2011.
- [WZJ<sup>+</sup>11] J. Wen, T. Zheng, W. Jiang, X. Deng, e Y. Fan, *A comparative study of helical-type and traditional-type artery bypass grafts: numerical simulation*, American Society of Artificial Internal Organs Journal **57** (2011), no. 5, 399–406.
- [Yan01] D. Yang, *C++ and object-oriented numeric computing for scientists and engineers*, Springer-Verlag New York, 2001.
- [ZDP<sup>+</sup>09] P. Zunino, C. D’Angelo, L. Petrini, C. Vergara, C. Capelli, e F. Migliavacca, *Numerical simulation of drug eluting coronary stents: mechanics, fluid dynamics and drug release*, Computer Methods in Applied Mechanics and Engineering **198** (2009), no. 45-46, 3633–3644.